Interpretable and fast dimension reduction of multivariate data

Thesis

Interpretable and Fast Dimension Reduction of Multivariate Data

by

Doyo Gragn Enki

B.Sc. (Statistics), M.Sc. (Statistics, Biostatistics)

A thesis submitted to The Open University
in fulfilment of the requirements for the degree of
Doctor of Philosophy (PhD)
in Statistics

Department of Mathematics and Statistics
Faculty of Mathematics, Computing & Technology
The Open University
Walton Hall, Milton Keynes, MK7 6AA, United Kingdom

November 2010

DATE OF SUBMISSION: 24 NOV 2010

DATE OF AWARD: 21 FEB 2011

Abstract

The main objective of this thesis is to propose new techniques to simplify the interpretation of newly formed 'variables' or components, while reducing the dimensionality of multivariate data. Most attention is given to the interpretation of principal components, although one chapter is devoted to that of factors in factor analysis. Sparse principal components are proposed, in which some of the component loadings are made exactly zero. One approach is to make use of the idea of correlation biplots, where orthogonal matrix of sparse loadings is obtained from computing the biplot factors of the product of principal component loading matrix and functions of their variances. Other approaches involve clustering of variables as a pre-processing step, so that sparse components are computed from the data or correlation matrix of each cluster. New clustering techniques are proposed for this purpose. In addition, a penalized varimax approach is proposed for simplifying the interpretation of factors in factor analysis, especially for factor solutions with considerably different sum of squares. This is done by adding a penalty term to the ordinary varimax criterion.

Data sets of varying sizes, both synthetic and real, are used to illustrate the proposed methods, and the results are compared with those of existing ones. In the case of principal component analysis, the resulting sparse components are found to be more

interpretable (sparser) and explain higher cumulative percentage of adjusted variance compared to their counterparts from other techniques. The penalized varimax approach contributes in finding a factor solution with simple structures which are not revealed by the standard varimax solution.

The proposed methods are very simple to understand and involve fast algorithms compared to some of the existing methods. They contribute much to the interpretation of components in a reduced dimension while dealing with dimensionality reduction of multivariate data.

Dedicated to my wife

and

the memory of my mother.

Acknowledgements

Thanks be unto God for His unspeakable gift.

I owe my deepest gratitude to Dr Nickolay Trendafilov, my principal supervisor, who played a major role from the initial to the final level of my study. This thesis would not have been possible without his guidance, encouragement, and support. I am very grateful to Professor Frank Critchley, my second supervisor, who read through the whole thesis and gave crucial comments and suggestions which improved the contents and structure of the thesis.

It is an honor for me to get advices and supports from the staffs in Statistics group, The Open University, a friendly environment which I will never forget. My special thanks go to Prof Paddy Farrington, Prof Paul Garthwaite, Prof John Gower, Prof Chris Jones, Dr Karen Vines, Dr Steffen Unkel, Dr Karim Anaya, and Dr Swarup De. I would like to thank my fellow-students, Angela Noufaily, Fadlala Elfadaly, and Osvaldo Anacleto-Junior, who have made available their support in a number of ways. I would like to show my gratitude to Tracy Johns, Sarah Frain, and Sara Griffin, for their help in the administrative wing. I am very grateful to Prof Ian Jolliffe and Dr Karen Vines for their willingness to examine this thesis, and Heather Whitaker for organizing and chairing the examination panel.

I am indebted to my wife, Mestawet (meaning 'Mirror' in Amharic), who stood by

my side all the time. Her encouragement, advice and taking over of my role in the household, especially during the last few months of thesis writing, were paramount.

I would like to thank all the people who provided their assistance to me in a form of advice, suggestions, and any other. My sincere thanks go to Dr Sharon Goodman, Dr Seife Ayele, Dr Yoseph Araya, Dr Abel Tilahun, Dr Fetene Bekele, Dr Wendimagegn Ghidey, Adane Fekadu, Mulugeta Degefa, Adane Girma, and Paulos Desalegn.

Last but not least, I am grateful to my and my wife's parents, whose supports were with me all the time. Above all, I thank the Research School of the Open University, the basis for all the works to come true.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and preliminaries

1.1 Introduction

When conducting an experiment or observing physical or social phenomenon, one usually makes records or measurements on a number of variables on specific observational units. The number of variables depends on the objective of the study, the characteristics of the individual (item or subject) under investigation and so on. An investigator will often include as many variables as possible in order not to miss relevant information in the future. As a result, most data sets are high-dimensional. Furthermore, such data sets are often characterized by the fact that the measurements are simultaneously taken from highly correlated variables, and a large number of variables conveys information that can be conveyed by only few original variables or linear combination of them.

The majority of data sets collected and/or analyzed by researchers in all fields of application are multivariate. Sometimes, it may make sense to deal with each variable separately, but in the majority of the cases, all or most of the variables are dealt with

simultaneously in order to get the maximum possible information. This leads to the need for multivariate data analysis. Data sets can include measurements from both qualitative and quantitative variables but, in this thesis, we are concerned with the data sets from quantitative variables.

An n-by-p data matrix $\mathbf{X}$ is viewed as a collection of n points or observations in a p-dimensional space. In most cases $n > p$ but, in many contemporary applications, the number of variables is comparable or even much larger than the number of observations. Such high-dimensional multivariate data sets may create problems in computational time, storage (memory), interpretation of results, visualizing data structures and so on.

A general approach to dealing with a high-dimensional multivariate data set is to reduce its dimension to a manageable size, say k ($< p$), while keeping as much of the original information as possible. There are two main approaches to do so – taking a subset of k variables or replacing the p original variables by k linear combinations of the variables (thereby forming new 'variables'). Several dimension-reducing techniques employing the latter approach are already available. The most efficient and well-known one is principal component analysis (PCA).

If $k \ll p$, then the reduction of dimensionality alone may justify the use of PCA. However, the technique is especially useful if the principal components (PCs) are readily interpreted. Unfortunately, the PCs are not always easily interpretable, especially for those involving a large number of original variables, as each principal component consists of a linear combination of all the original variables with nonzero-loadings. The classical way of ignoring loadings whose absolute values are below some specified threshold while interpreting a PC is found to be misleading. As a result, several

approaches have been proposed for simplifying interpretation.

Modern simplifying methods propose sparse principal components, in which many of the loadings of a component are forced to be exactly zero. This can be done by either restricting the coefficients of the variables to only a few integer values, or imposing a certain optimization criterion which drives some of the component loadings to zero. The objective of such methods is to approximate the PCs in such a way that they are simpler to interpret, without sacrificing much variance.

However, sparse components based on existing simplifying methods are either not sparse enough or their interpretation is not much simpler than the original components. Another difficulty is the adjustment of certain tuning parameter(s) which may be subjective or time consuming, due to requiring cross-validation. This thesis contributes further to the interpretation of dimension-reducing techniques, especially PCA, by proposing simple and fast methods of constructing sparse components. Each of the resulting sparse components has, typically, a higher number of zero-loadings than those based on existing methods, with a minimal loss of information. Moreover, these sparse components are non-overlapping with each other with respect to the variables involved, leading to simpler interpretation.

1.2 Notation

Some specific notations are uniquely defined in each chapter, but some of them are globally used throughout the document. Unless explicitly stated otherwise in a particular chapter, n denotes the number of observations (samples) while p denotes the number of variables. In general, bold small letters refer to vectors while bold capital

letters refer to matrices. A p-vector of variables is denoted by $\mathbf{x}$, while $\mathbf{X} = (x_{ij})$ denotes an n-by-p data matrix, where x_{ij} represents the value of the ith observation on the jth variable. The transposes of $\mathbf{x}$ and $\mathbf{X}$ are denoted by $\mathbf{x}^\top$ and $\mathbf{X}^\top$, respectively. Similarly, a vector of constants is denoted by a bold small letter, say $\mathbf{a}$, and a matrix of constants by bold capital letter, say $\mathbf{A}$. A matrix $\mathbf{A}$ of k ($\leq p$) columns is sometimes written as $\mathbf{A}_k$. In each chapter, non-bold capital letters may be used for other purposes. For instance, the variance of $\mathbf{x}$ is written in short as $V(\mathbf{x})$.

The covariance matrix of $\mathbf{x}$ is denoted by $\mathbf{\Sigma} = (\sigma_{ij})$, whose (i,j)th element σ_{ij} is the covariance between the ith and the jth elements of $\mathbf{x}$ when $i \neq j$, and the variance of the ith element of $\mathbf{x}$ when $i = j$. Similarly, the correlation matrix is denoted by $\mathbf{R} = (r_{ij})$, with r_{ij} denoting the (i,j)th element of $\mathbf{R}$. For simplicity, the mean of each element of $\mathbf{x}$ is assumed to be zero.

A p-dimensional real-space is denoted by $\mathbb{R}^p$. Sometimes, the dimension of a vector or a matrix is given by a subscript. Thus, $\mathbf{X}_{n\times p}$ denotes a matrix $\mathbf{X}$ of dimension n-by-p and $\mathbf{x}_p$ denotes a vector $\mathbf{x}$ of dimension p-by-1. The identity matrix of dimension p-by-p is denoted by $\mathbf{I}_p$. Vectors of ones and zeros are denoted as $\mathbf{1}$ and $\mathbf{0}$, respectively. The determinant of a square matrix $\mathbf{\Lambda}$ is written as $\det(\mathbf{\Lambda})$, and its trace as $\mathrm{trace}(\mathbf{\Lambda})$. The diagonal elements of $\mathbf{\Lambda}$ are written as $\mathrm{diag}(\mathbf{\Lambda})$. However, if $\boldsymbol{\lambda}$ is a vector of elements $(\lambda_1, \lambda_2, \ldots, \lambda_k)$, then $\mathrm{diag}(\boldsymbol{\lambda})$ denotes a diagonal matrix. The inverse of $\mathbf{\Lambda}$ is denoted by $\mathbf{\Lambda}^{-1}$.

Random variables and their realizations are not differentiated in this thesis. In addition, most discussions will not distinguish between samples and populations, with $\mathbf{\Sigma}$ and $\mathbf{R}$ referring to either population or sample.

1.3 Outline of the thesis

Each of the chapters in this thesis can be read as a self-contained article. In general, the thesis is organized as follows. Chapter 2 deals with common statistical techniques used in dimensionality reduction. It is intended to give a general overview of the techniques, and is by no means exhaustive. The chapter begins with a brief introduction to principal component analysis (PCA), the most efficient and well-known method. This is followed by a related but distinct technique, factor analysis. Other techniques briefly discussed in this chapter include linear discriminant analysis, cluster analysis, canonical correlation analysis, multidimensional scaling and biplots.

A literature review of some of the interpretable dimension reduction approaches is given in Chapter 3, which mainly targets on the interpretation of PCs. The review starts with the simple structure rotation technique, discussed with respect to the commonly used varimax rotation criterion. However, the majority of the chapter deals with the more recent simplifying approaches, which involve constraining the loadings of the components. The remaining part of the chapter deals with subset selection, which is concerned with the selection of subsets of variables, in contrast to linear combinations of variables such as PCs.

Chapter 4 proposes a new simple method of deriving sparse PCs. It uses an idea from correlation biplots, and so is called sparse biplots (sBarse) component analysis. The method uses as input the loadings and variances of PCs, and produces simplified loadings for all components simultaneously. The advantages and disadvantages of the approach are also discussed.

Another approach proposed for simplifying the interpretation of PCs is based on

clustering of variables. Chapter 5 deals with interpretable PCs where each sparse (interpretable) PC is constructed from the data matrix of a cluster of variables. For this purpose, a new clustering method, called weighted-variance, is proposed. The resulting sparse PCs are also compared with those based on existing clustering methods. This method is designed especially for the case where the number of samples (n) exceeds the number of variables (p).

Chapter 6 extends the cluster-based sparse PC approach to the general case with $p \gg n$ or $n > p$, based on another new method of clustering variables, called semi-partition. It is designed especially for microarray gene expression data sets where the number of genes (variables) is far larger than the number of samples.

Unlike the preceding chapters, where the main concern is the interpretation of PCs, Chapter 7 proposes an approach for facilitating the interpretation of factor loadings in factor analysis. It contributes to the varimax rotation problem, by introducing an additional penalty constraint to the original criterion.

The thesis ends with a short discussion and summary in Chapter 8, where each chapter is briefly summarized. Some future research directions are also indicated in this chapter.

The methods proposed in each of Chapters 4 to 7 are applied to different kinds of data sets. These include synthetic as well as real data sets of varying dimension. The real data sets involve both cases of $n > p$ and $p \gg n$. The majority of the data analysis is performed by `MATLAB` programs (MATLAB, 2009), which are written by and available with the author. A few programs in `R` are also used as a supplement.

Chapter 2

Dimension reduction in multivariate data analysis

In this chapter, we briefly review some statistical methods which (directly or indirectly) involve dimension reduction of high-dimensional multivariate data. Sections 2.1 to 2.7, respectively, give a brief review of principal component analysis, factor analysis, linear discriminant analysis, cluster analysis, canonical correlation analysis, multidimensional scaling and biplots.

2.1 Principal component analysis

Principal component analysis (PCA) is the most popular and efficient technique for reducing the dimension of a high-dimensional multivariate data. It takes observations on p correlated variables and transforms them into new uncorrelated variables, called principal components (PCs), which successively account for as much variation as possible in the original variables (Jolliffe, 2002; Krzanowski, 1988; Rao, 1964). The term 'principal component' was first introduced by Hotelling (1933).

Consider a p-vector of random variables $\mathbf{x}$ with a known covariance matrix $\mathbf{\Sigma}$. [A similar procedure can be followed if the correlation matrix $\mathbf{R}$ is used instead of $\mathbf{\Sigma}$.] Then, PCA aims to find the linear combinations

$$y_i = \mathbf{a}_i^\top \mathbf{x}, \ i = 1, 2, \ldots, p, \tag{2.1}$$

which successively maximize the variance

$$\mathrm{V}(y_i) = \mathbf{a}_i^\top \mathbf{\Sigma} \mathbf{a}_i$$

subject to the constraints

$$\mathbf{a}_i^\top \mathbf{a}_i = 1 \text{ and } \mathbf{a}_i^\top \mathbf{a}_j = 0, \ i < j,$$

where $\mathbf{a}_i$ is a p-vector of constants $a_{i1}, a_{i2}, \ldots, a_{ip}$. It has been shown (Jolliffe, 2002) that $\mathrm{V}(y_i)$ is maximized when $\mathbf{a}_i$ is the ith eigenvector corresponding to the ith largest eigenvalue λ_i of $\mathbf{\Sigma}$. The random variable y_i gives the ith principal component of $\mathbf{x}$, with the property that the $\mathbf{a}_i$'s are orthonormal and y_i is uncorrelated to y_j for any $i \neq j$. Furthermore, λ_i is the variance of the ith PC. It is assumed here that all the eigenvalues are distinct, but, theoretically, eigenvalues can be equal. Some problems related to PCs with equal variances are discussed in Sections 2.4 and 3.4 of Jolliffe (2002).

The PCs can also be expressed in matrix form. If $\mathbf{A} = (\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_p)$ denotes an orthogonal matrix of eigenvectors, and $\mathbf{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$ denotes a diagonal matrix of the corresponding eigenvalues with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \ (\geq 0)$, then the vector $\mathbf{y}$ of PCs with the ith element y_i is given as

$$\mathbf{y} = \mathbf{A}^\top \mathbf{x}, \tag{2.2}$$

subject to $\mathbf{A}^\top\mathbf{A} = \mathbf{A}\mathbf{A}^\top = \mathbf{I}_p$. Note that the diagonal matrix of variances of the PCs can be given as

$$\mathbf{\Lambda} = \mathbf{A}^\top\mathbf{\Sigma}\mathbf{A}. \tag{2.3}$$

The orthogonality of $\mathbf{A}$ leads to an alternative expression for (2.3) as

$$\mathbf{\Sigma} = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^\top, \tag{2.4}$$

often called the spectral decomposition.

The criterion used to find PCs in the above procedure is called variance maximization. Principal components can also be obtained using the singular value decomposition (SVD) approach. For a mean-centered n-by-p data matrix $\mathbf{X}$ of rank r ($\leq p$), the SVD is

$$\mathbf{X} = \mathbf{U}\mathbf{L}\mathbf{A}^\top$$

where $\mathbf{U}$ and $\mathbf{A}$ are, respectively, n-by-r and p-by-r orthogonal matrices and $\mathbf{L}$ is r-by-r diagonal matrix of singular values.Thus, $\mathbf{A}$ gives the eigenvectors of the covariance matrix $\mathbf{X}^\top\mathbf{X}$ (and hence the loadings of the PCs), while the diagonal elements of $\mathbf{L}$ give the square roots of the corresponding eigenvalues. The matrix of PC scores $\mathbf{Y}$ can be derived from $\mathbf{U}$ and $\mathbf{L}$ as $\mathbf{Y} := \mathbf{U}\mathbf{L}$.

Furthermore, $\mathbf{X}^\top\mathbf{X}$ and $\mathbf{X}\mathbf{X}^\top$ are both symmetric and have the same non-zero eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_r$. The columns of $\mathbf{U}$ give the eigenvectors of $\mathbf{X}\mathbf{X}^\top$ corresponding to the nonzero eigenvalues. As given by Rao (1964), if $\mathbf{a}_i$ and $\mathbf{u}_i$ are the columns of $\mathbf{A}$ and $\mathbf{U}$, respectively, corresponding to the ith eigenvalue λ_i, then $\mathbf{a}_i = \lambda_i^{-1/2}\mathbf{X}^\top\mathbf{u}_i$ and $\mathbf{u}_i = \lambda_i^{-1/2}\mathbf{X}\mathbf{a}_i$ for $i = 1, 2, \ldots, r$.

The orthogonality of the matrix of loadings and the uncorrelatedness of the components are the two properties that make PCs so attractive for application. Despite

these nice properties, Gower (1967) discusses some critiques of PCA. The first critique is related to the scaling on which the variables were measured. There is no problem with regular PCA if all variables are of same type (e.g. lengths) measured on the same scale (e.g. cm). But, if the variables are measured on different units, then a change in the scales will lead to different PCs. Therefore, each variable should be standardized to a dimensionless quantity (for instance, dividing by its standard deviation) so that the sum of squares and cross products matrix $\mathbf{X}^\top\mathbf{X}$ becomes the correlation matrix. The other critique addresses the fact that the sum of squares of the loadings for the ith PC should be unity. In the absence of this restriction, the variance of y_i can be made as large as we want by simply enlarging the loadings. These and other more critiques are detailed in Gower (1967).

In practice, the first k ($\ll p$) PCs usually account for most of the variation in the original p variables, and hence the original data set can be reduced to a set consisting of n measurements on k principal components (hence reducing dimensionality). There are a number of techniques suggested for choosing the number k of principal components to retain. The rule constructed by Kaiser (1960) (based on the correlation matrix) suggests that any PC with variance $\lambda_i < 1$ shouldn't be retained as it contains less information than one of the original variables. However, Jolliffe (1972) argues that $\lambda_i = 1$ as cut-off level retains too few components, and hence suggested to use $\lambda_i = 0.7$. Another technique discussed in Cattell (1966) is to use a scree plot, where the number of principal components to retain is inferred from the 'elbow' of the scree graph. Alternatively, one can use the cumulative percentage of total variation, which suggests to retain the smallest number of principal components whose cumulative contribution of variation $\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{p}\lambda_i}$ exceeds .80 or more. More discussion on the approaches to the

estimation of k is given in Jolliffe (2002).

A geometrical interpretation of PCs is given as follows (Gower, 1967; Pearson, 1901). Let the sample values of the n points in a p-dimensional space be given as

$$\mathbf{X} = \begin{pmatrix} x_{11} & \cdots & x_{1p} \\ \cdots & \cdots & \cdots \\ x_{n1} & \cdots & x_{np} \end{pmatrix},$$

where each row vector represents a point. Denote the ith point by P_i $(i = 1, \ldots, n)$. For simplification, assume that every variate is centered, i.e., the origin of the n points P_i is at the centroid (or center of gravity) $\overline{P}$. The total sum of squares of the sample points from the centroid is $\sum_{i=1}^{n}(\overline{P}P_i)^2$. If Q_i is the projection of P_i onto the first eigenvector of the sum of squares and products matrix $\mathbf{X}^\top\mathbf{X}$, then $\sum(\overline{P}P_i)^2 = \sum(P_iQ_i)^2 + \sum(\overline{P}Q_i)^2$ using Pythagoras' theorem. As the total sum of squares $\sum(\overline{P}P_i)^2$ is fixed and $\sum(P_iQ_i)^2$ needs to be minimized, then $\sum(\overline{P}Q_i)^2$ must be maximized. This last term gives the variance of the linear combination with coefficients of the eigenvector corresponding to the maximal eigenvalue of $\mathbf{X}^\top\mathbf{X}$ (see also Krzanowski (1988)). Thus, PCA identifies a line which gives the best fit to the n points. Such a line minimizes a criterion involving the sum of the squares of the perpendicular distances from each of the points P_i onto the line. Such best-fit line passes through the centroid and its direction cosines are one of the eigenvectors of $\mathbf{\Sigma} = \mathbf{X}^\top\mathbf{X}$.

2.2 Factor analysis

Factor analysis (FA) is another technique that involves dimension reduction. The essence of FA is that a set of p observed random variables $\mathbf{x} = (x_1, x_2, \ldots, x_p)^\top$ are

expressed (with error) as a linear function of k ($\ll p$) hypothetical (latent) random variables called *common factors*. Let $\mathbf{f} = (f_1, f_2, \ldots, f_k)^\top$ denotes the vector of common factors. Then, the factor model is expressed as

$$\mathbf{x} = \mathbf{\Gamma f} + \boldsymbol{\epsilon}, \tag{2.5}$$

where $\mathbf{\Gamma}$ is a p-by-k (constant, but unknown) matrix of factor loadings, and $\boldsymbol{\epsilon}$ is a p-by-1 random vector of errors or *specific factors*. The commonly used assumptions of the factor model are

$$E(\mathbf{x}) = E(\mathbf{f}) = E(\boldsymbol{\epsilon}) = \mathbf{0},$$

and

$$E(\mathbf{ff}^\top) = \mathbf{I}_k, \quad E(\mathbf{f}\boldsymbol{\epsilon}^\top) = \mathbf{0}, \quad \text{and} \quad E(\boldsymbol{\epsilon}\boldsymbol{\epsilon}^\top) = \mathbf{\Psi},$$

where $E(.)$ stands for "expected value", and $\mathbf{\Psi}$ is a diagonal matrix of elements $\psi_1, \psi_2, \ldots, \psi_p$. That is, the common factors are uncorrelated with each other and are of unit variance, and the error terms are also uncorrelated with each other and of the common factors. Thus, the covariance matrix $\mathbf{\Psi}$ of the error terms is diagonal. Taking these assumptions into consideration, the covariance matrix of $\mathbf{x}$ is modeled by

$$\mathbf{\Sigma} = E(\mathbf{xx}^\top) = \mathbf{\Gamma\Gamma}^\top + \mathbf{\Psi}. \tag{2.6}$$

Both $\mathbf{\Gamma}$ and $\mathbf{\Psi}$ in (2.6) are unknown parameters to be estimated from experimental data. There are different methods of estimating the parameters $\mathbf{\Gamma}$ and $\mathbf{\Psi}$ (see, for instance, Lawley and Maxwell (1971) and Mulaik (1972)).

Given the matrix $\mathbf{\Sigma}$ and assuming that $\mathbf{\Psi}$ is uniquely defined with positive diagonal elements, then $\mathbf{\Sigma} - \mathbf{\Psi}$ (also called the reduced covariance matrix) is a rank-k covariance

matrix of $\mathbf{x}$, where each diagonal element represents the part of variance due to the k common factors. This is called the communality of the variate (Lawley and Maxwell, 1971).

In deriving the observed variables as a linear combination of the common and the specific factors, the matrix of factor loadings $\mathbf{\Gamma}$ gives weights assigned to the common factors. If all the factors are assumed to be uncorrelated to one another, then $\mathbf{\Gamma}$ is equivalent to the matrix of correlation between the common factors and the observed variables. In fact, if $\mathbf{R}_{fx}$ represents the matrix of correlations between $\mathbf{f}$ and $\mathbf{x}$ each with unit variance, then the assumptions of the factor model leads to

$$\mathbf{R}_{fx} = E(\mathbf{x}\mathbf{f}^\top) = E[(\mathbf{\Gamma}\mathbf{f} + \epsilon)\mathbf{f}^\top] = \mathbf{\Gamma}E(\mathbf{f}\mathbf{f}^\top) = \mathbf{\Gamma}.$$

In this sense, a larger element γ_{ij} of $\mathbf{\Gamma}$ corresponds to a high correlation between the ith observed variable and the jth common factor.

Some authors consider PCA as a special case of FA, but the two are quite distinct techniques (Jolliffe, 2002). Both FA and PCA use covariance or correlation matrix of variables, but with different aims. PCA gives more attention to the diagonal (variances) of the matrix, while FA gives more attention to the off-diagonal (covariance or correlation) values. No hypothesis or assumption needs to be made about the variables in PCA, while FA is based on a model with particular assumptions about the parameters. In addition, PCs are linear combinations of the original variables while in FA, the original variables are linear combinations of hypothetical variates or factors.

Despite the differences between the two, FA is frequently used as an alternative to PCA for reducing the dimension of large data sets (Krzanowski, 1988, Sec 16.2.8). It reduces the p manifest variables to a relatively small number k of uncorrelated common

factors assuming that the FA model holds, which is usually left unchecked.

2.3 Linear discriminant analysis

Discriminant analysis is an exploratory multivariate technique which allows the researcher to study the difference between two or more existing groups (or populations or classes) of observations by constructing discriminant functions or rules that discriminate between the groups best.

Suppose we have a random sample $\mathbf{x}_1, \mathbf{x}_1, \ldots, \mathbf{x}_n$ of n p-dimensional observations from a population with probability density function $f(\mathbf{x})$. Let the observations be divided into g groups *a priori*, say $G_1, G_2, \ldots, G_g$ ($g \geq 2$), each containing n_i observations, with $\sum_{i=1}^{g} n_i = n$. Assume that with each group there is an associated probability density function $f_i(\mathbf{x})$ on $\mathbb{R}^p$, $i = 1, 2, \ldots, g$. Given that an object is known to come from one of the g groups G_i, the aim is to allocate the object to this group on the basis of p measured characteristics $\mathbf{x}$ associated with the object. The allocation requires a discriminant (or allocation) rule, which also requires dividing $\mathbb{R}^p$ into g disjoint regions $R_1, R_2, \ldots, R_g$. Then the discriminant rule is to allocate $\mathbf{x}$ to G_i if $\mathbf{x} \in R_i$. When the groups are known, the discrimination can be made either using the maximum likelihood discriminant rule or the Bayes discriminant rule, under additional distributional assumptions (Mardia *et al.*, 1982, Chapter 11).

Fisher's linear discriminant analysis (LDA) looks for a discrimination rule without involving any distributional assumption about the population. Let $\mathbf{x}_{ij}$ denotes the jth individual from the ith group. The sample mean vector for the ith group is given by $\bar{\mathbf{x}}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_{ij}$ and the overall mean vector is given by $\bar{\mathbf{x}} = \frac{1}{n} \sum_{i=1}^{g} n_i \bar{\mathbf{x}}_i$. Assuming

the same covariance matrix in each group, the pooled within-group scatter matrix is defined as

$$\mathbf{S}_W = \sum_{i=1}^{g}\sum_{j=1}^{n_i}(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)(\mathbf{x}_{ij} - \bar{\mathbf{x}}_i)^\top, \tag{2.7}$$

and the between-group scatter matrix is defined as

$$\mathbf{S}_B = \sum_{i=1}^{g} n_i(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})(\bar{\mathbf{x}}_i - \bar{\mathbf{x}})^\top. \tag{2.8}$$

Fisher suggested using the ratio of the between-groups sum-of-squares ($\mathbf{S}_B$) to within-groups sum-of-squares ($\mathbf{S}_W$) to determine the degree of separation between the groups. However, we can reduce the multivariate observations $\mathbf{x}_{ij}$ to univariate observations $y_{ij} = \boldsymbol{\omega}^\top \mathbf{x}_{ij}$ and compute the usual sums of squares:

$$\sum_i \sum_j (y_{ij} - \bar{y}_i)^2 = \boldsymbol{\omega}^\top \mathbf{S}_W \boldsymbol{\omega}$$

and

$$\sum_i \sum_j (\bar{y}_i - \bar{y})^2 = \boldsymbol{\omega}^\top \mathbf{S}_B \boldsymbol{\omega}.$$

Now the first step in LDA is finding a transformation vector $\boldsymbol{\omega}$ so that the ratio

$$\frac{\boldsymbol{\omega}^\top \mathbf{S}_B \boldsymbol{\omega}}{\boldsymbol{\omega}^\top \mathbf{S}_W \boldsymbol{\omega}} \tag{2.9}$$

is maximized. This can also be equivalently given by a generalized eigenvalue problem:

$$(\mathbf{S}_B - \lambda \mathbf{S}_W)\boldsymbol{\omega} = \mathbf{0}$$

or

$$(\mathbf{S}_W^{-1}\mathbf{S}_B - \lambda \mathbf{I})\boldsymbol{\omega} = \mathbf{0}$$

Then, λ must be the largest eigenvalue λ_1 of $\mathbf{S}_W^{-1}\mathbf{S}_B$ with the corresponding eigenvector $\boldsymbol{\omega} = \boldsymbol{\omega}_1$. That means, $\boldsymbol{\omega}_1$ gives the direction in the p-dimensional data

space along which the between-group variability is greatest relative to the within-group variability. Similar procedures can be used to obtain the remaining directions $\boldsymbol{\omega}_2, \boldsymbol{\omega}_3, \ldots, \boldsymbol{\omega}_p$ (eigenvectors of $\mathbf{S}_W^{-1}\mathbf{S}_B$) corresponding to the eigenvalues $\lambda_2, \lambda_3, \ldots, \lambda_p$, by successively maximizing (2.9) subject to having uncorrelated new variables. Let $\boldsymbol{\Omega} = (\boldsymbol{\omega}_1, \boldsymbol{\omega}_2, \ldots, \boldsymbol{\omega}_k)$ with $k \leq \min(p, g-1)$ and consider the $\mathbf{y}_{ij} = \boldsymbol{\Omega}\mathbf{x}_{ij}$. Then the first k elements of $\mathbf{y}_{ij}$ are the first k discriminant coordinates (Seber, 2004).

In this sense, LDA provides a low-dimensional representation of a data matrix such that the true differences between the groups in the original space are reproduced as accurately as possible (Krzanowski, 1988). The matrix $\mathbf{Y} = \mathbf{X}\boldsymbol{\Omega}$ is a linear transformation of $\mathbf{X}$ into a new $n \times k$ data space $\mathbf{Y}$ and $\boldsymbol{\Omega}$ is the transformation matrix that makes the groups to be best separated in the new space $\mathbf{Y}$ with respect to the criterion (2.9). Usually k is required to be considerably smaller than p, say $k = 2$. However, unlike PCA where the principal component loadings $\mathbf{A}$ are orthogonal in the original space, $\mathbf{A}^\top\mathbf{A} = \mathbf{I}$, the discriminant variate loadings are orthogonal in the $\mathbf{S}_W$-space, $\boldsymbol{\Omega}^\top\mathbf{S}_W\boldsymbol{\Omega} = \mathbf{I}$, which gives non-orthogonal projection in the original space.

2.4 Cluster analysis

Cluster analysis is a multivariate technique which groups objects (variables or items) into an *unknown* number of clusters based on certain measures of similarity or dissimilarity (Everitt, 1974; Hartigan, 1975; Späth, 1980; Romesburg, 2004; Seber, 2004). The main goal of cluster analysis is to search the data for 'natural' groupings of the objects, so that objects within the same group are more homogeneous. That is, it groups objects into clusters such that pairs of objects from the same cluster are more

similar to each other than are pairs of objects from different clusters. Such homogeneity may help to identically treat the objects in the same group for the purpose of some further analysis, compared to the whole heterogeneous data set. The method can also be used in the absence of a clear-cut group structure in the data, to separate a set of objects into constituent groups so that members of any group differ from one another as little as possible based on a given criterion.

Clustering techniques can be broadly divided into two as hierarchical and non-hierarchical (partitioning). In the hierarchical clustering technique, the clusters are themselves classified into groups, the process being repeated at different levels to form a 'cluster tree'. This technique is characterized by either a series of successive merging or successive divisions, leading, respectively, to agglomerative (bottom up) and divisive (top down) hierarchical methods. The agglomerative hierarchical method starts with as many groups as objects, and a pair of groups are successively fused together until a single group consisting of all the objects is formed. The divisive hierarchical method works in the opposite direction: it starts with a single group consisting of all the objects, and each group is successively divided into two groups until each object forms a group. In the non-hierarchical clustering techniques, the objects are split into overlapping or non-overlapping clusters.

Clustering requires to define a measure of similarity or dissimilarity (distance) between each pair of objects in order to produce a simple group structure from a complex data set. If items are to be clustered, the proximity measure is usually given by some sort of distance, whereas variables are usually grouped on the basis of similarity measures, such as correlation coefficients or measures of association.

Suppose $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{ip}]^\top$ and $\mathbf{x}_{i'} = [x_{i'1}, x_{i'2}, \ldots, x_{i'p}]^\top$ are two vectors of

observations (or two points in a p-dimensional space), which correspond to two objects described by the rows of $\mathbf{X}$. The most common dissimilarity measure for measuring the nearness of the two points is the Euclidean distance. From the general L_2-norm of a vector

$$\|\mathbf{x}_i\|_2 = \left[\sum_{l=1}^{p} x_{il}^2\right]^{\frac{1}{2}} = \sqrt{\mathbf{x}_i^\top \mathbf{x}_i},$$

the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{x}_{i'}$ is

$$\|\mathbf{x}_i - \mathbf{x}_{i'}\|_2 = \sqrt{(\mathbf{x}_i - \mathbf{x}_{i'})^\top (\mathbf{x}_i - \mathbf{x}_{i'})}.$$

A number of different dissimilarity measures have been proposed in the literature. In the agglomerative hierarchical clustering technique, the most common measures are the single linkage (minimum distance or nearest neighbour), the complete linkage (maximum distance or furthest neighbour), and the average linkage (average distance). Let $\mathbf{D} = (d_{ij})$ denotes the n-by-n symmetric matrix of dissimilarities and C_1 and C_2 denote two clusters. In the single linkage context, the distance between C_1 and C_2 is given by the smallest dissimilarity between a member of C_1 and a member of C_2; that is,

$$d_{(C_1)(C_2)} = \min\{d_{uv} : u \in C_1, v \in C_2\},$$

while for the complete linkage method the distance is given by

$$d_{(C_1)(C_2)} = \max\{d_{uv} : u \in C_1, v \in C_2\}.$$

In the average linkage method, the distance between two clusters is defined by the average distance between all pairs of items where one member of a pair belongs to each cluster. Mathematically, this is given by

$$d_{(C_1)(C_2)} = \frac{1}{n_1 n_2} \sum_{u \in C_1} \sum_{v \in C_2} d_{uv},$$

where n_1 and n_2 denote the number of objects in C_1 and C_2, respectively.

Most clustering techniques are designed for grouping observations rather than variables (Friedman and Rubin, 1967). However, we are interested in clustering variables, where the correlation coefficients between the variables are the natural similarities. Pairs of variables with relatively large correlations are considered to be 'close' to each other, while pairs of variables with relatively small correlations are considered to be 'far away' from each other. Thus, each cluster usually contains highly correlated variables, with each variable corresponding to one and only one cluster; i.e., the clusters are assumed to be non-overlapping with respect to the variables.

Consider a p-vector $\mathbf{x}$ of variables with correlation matrix $\mathbf{R}$. The following algorithm summarizes the hierarchical linkage method for clustering variables.

1. Start with p clusters, each containing a single variable.

2. Search the matrix $\mathbf{R}$ for the most correlated (least dissimilar) pair of clusters. Let these clusters be I and J with correlation coefficient r_{IJ}.

3. Merge clusters I and J and label the newly formed cluster as IJ. Update the entries in the correlation matrix by first deleting the rows and columns corresponding to clusters I and J, and then adding a row and column giving the 'correlations' between cluster IJ and the remaining clusters.

4. Repeat steps 2 and 3 a total of $p-1$ times. Record the identity of clusters that are merged and the coefficients at which the mergers take place.

In step 3, the merging of two clusters is based on one of the distance measures (single, complete, or average linkages), but using r_{ij} instead of d_{ij}, which switches `Max` and `Min` in single and complete link.

The most popular non-hierarchical clustering method is the k-means method. It starts by partitioning the items into k ($< n$) initial clusters (where k is fixed *a priori*), and proceeds with re-assigning each item to a cluster whose centroid (mean) is nearest, until no more reassignments take place. The method can also be adapted to grouping the p original variables into k ($< p$) clusters.

Dimension reduction in cluster analysis can be related to the notion of variable selection. If we wish to reduce the number of variables without sacrificing much information, then the variables are first grouped into non-overlapping clusters and then one variable is retained from each cluster. In addition, cluster analysis is connected with PCA in that for well-defined clusters there is one high-variance PC and one or more low-variance PCs associated with each cluster (Jolliffe, 2002, pp. 213).

2.5 Canonical correlation analysis

Suppose that a p-vector random variable $\mathbf{x}$ is divided into a p_1-vector $\mathbf{x}_1$ and a p_2-vector $\mathbf{x}_2$, where $p_1 + p_2 = p$. The objective of canonical correlation analysis is to identify the canonical correlation vectors $\boldsymbol{\alpha}_1$ and $\boldsymbol{\alpha}_2$ such that the correlation between the linear combinations (also called canonical variables) $\phi = \boldsymbol{\alpha}_1^\top \mathbf{x}_1$ and $\varphi = \boldsymbol{\alpha}_2^\top \mathbf{x}_2$ is maximized (Mardia *et al.*, 1982, Chap 10).

Assume $\mathbf{x}_1$ and $\mathbf{x}_2$ have means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$, and that the covariance matrix $\boldsymbol{\Sigma}$ of $\mathbf{x}$ is correspondingly partitioned as

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix},$$

where $\boldsymbol{\Sigma}_{ii}$ ($i = 1, 2$) is a $p_i \times p_i$ covariance matrix corresponding to $\mathbf{x}_i$ and $\boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21}^\top$.

Then, the squared correlation ϱ^2 between the two linear functions ϕ and φ is given by

$$\varrho^2 = \frac{(\boldsymbol{\alpha}_1^\top \boldsymbol{\Sigma}_{12}\boldsymbol{\alpha}_2)^2}{(\boldsymbol{\alpha}_1^\top \boldsymbol{\Sigma}_{11}\boldsymbol{\alpha}_1)(\boldsymbol{\alpha}_2^\top \boldsymbol{\Sigma}_{22}\boldsymbol{\alpha}_2)}.$$

Assuming that $\boldsymbol{\alpha}_1^\top \boldsymbol{\Sigma}_{11}\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_2^\top \boldsymbol{\Sigma}_{22}\boldsymbol{\alpha}_2 = 1$, it is found (Seber, 2004, Sec. 5.7) that the maximum value of ϱ^2, say ϱ_1^2, is the largest eigenvalue of $\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$ (equivalently, of $\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}$). This maximum value occurs when $\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_1^{(1)}$, the eigenvector of $\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$ corresponding to ϱ_1^2, and $\boldsymbol{\alpha}_2 = \boldsymbol{\alpha}_2^{(1)}$, the corresponding eigenvector of $\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}$. Here, $\sqrt{\varrho_1^2}$ is termed the first canonical correlation between $\mathbf{x}_1$ and $\mathbf{x}_2$, while $\phi^{(1)} = \boldsymbol{\alpha}_1^{(1)\top}\mathbf{x}_1$ and $\varphi^{(1)} = \boldsymbol{\alpha}_2^{(1)\top}\mathbf{x}_2$ are the first canonical variables. The second canonical variables, $\phi^{(2)} = \boldsymbol{\alpha}_1^{(2)\top}\mathbf{x}_1$ and $\varphi^{(2)} = \boldsymbol{\alpha}_2^{(2)\top}\mathbf{x}_2$, are chosen so that $\phi^{(2)}$ is uncorrelated with $\phi^{(1)}$, $\varphi^{(2)}$ is uncorrelated with $\varphi^{(1)}$, and $\phi^{(2)}$ and $\varphi^{(2)}$ have maximum squared correlation $\sqrt{\varrho_2^2}$. The procedure can be extended to choose the jth pair of eigenvalues and eigenvectors so that $\sqrt{\varrho_j^2}$ gives the jth maximum canonical correlation, and $\phi^{(j)} = \boldsymbol{\alpha}_1^{(j)\top}\mathbf{x}_1$ and $\varphi^{(j)} = \boldsymbol{\alpha}_2^{(j)\top}\mathbf{x}_2$ give the jth canonical variables, $j = 1, 2, \ldots, k$, with the constraints that $\phi^{(j)}$ is uncorrelated with $\phi^{(l)}$ $(l = 1, 2, \ldots, j-1)$ and $\varphi^{(j)}$ is uncorrelated with $\varphi^{(l)}$ $(l = 1, 2, \ldots, j-1)$. Thus, the procedure may help to reduce $\mathbf{x}_1$ and $\mathbf{x}_2$ to k-dimensional vectors $\boldsymbol{\phi}_k = (\phi^{(1)}, \phi^{(2)}, \ldots, \phi^{(k)})$ and $\boldsymbol{\varphi}_k = (\varphi^{(1)}, \varphi^{(2)}, \ldots, \varphi^{(k)})$, respectively.

2.6 Multidimensional scaling

Recall from Section 2.1 that PCA is a dimension-reducing technique which replaces the n p-dimensional vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ by n k-dimensional vectors (principal components) $\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n$, where k is much smaller than p. Multidimensional scaling (MDS) is another dimension-reducing technique which displays high-dimensional multivariate

data in a low-dimensional space. It uses the interpoint distances $d_{uv} = \| \mathbf{x}_u - \mathbf{x}_v \|$ between each pair of objects (u,v), and tries to find a set of k-dimensional vectors $\mathbf{y}_i$ with interpoint distances $d'_{uv} = \| \mathbf{y}_u - \mathbf{y}_v \|$ such that $d_{uv} \approx d'_{uv}$ for all u, v. The distances d_{uv} are often given by the proximity (similarity or dissimilarity) measures between pairs of objects. There are $n(n-1)/2$ such proximity measures available, which form the data set analyzed by MDS. Generally speaking, MDS covers any technique that produces a graphical representation of objects from multivariate data (Cox and Cox, 1994).

The 'classical' solution to MDS is as follows. Let $\mathbf{D} = (d_{uv})$ be a matrix of dissimilarities with $d_{uu} = 0$ and $d_{uv} = d_{vu} \geq 0$. Define a matrix $\mathbf{T} = (t_{uv})$ where $t_{uv} = -\frac{1}{2} d_{uv}^2$ and

$$\bar{\mathbf{T}} = \mathbf{M}_n^\top \mathbf{T} \mathbf{M}_n$$

where $\mathbf{M}_n = \mathbf{I}_n - (1/n)\mathbf{1}_n\mathbf{1}_n^\top$ is the centering matrix. The matrix $\mathbf{D}$ may or may not be Euclidean, but it is shown (Gower, 1966; Seber, 2004, p.236) that $\mathbf{D}$ is Euclidean if and only if $\bar{\mathbf{T}}$ is positive semidefinite. When $\mathbf{D}$ is Euclidean, the configuration $\mathbf{y}_i$ from the classical method of multidimensional scaling is closely connected with PCA. Once the matrix $\bar{\mathbf{T}}$ is obtained, the next step is to extract the k largest positive eigenvalues of $\bar{\mathbf{T}}$ with corresponding normalized eigenvectors $\mathbf{Y}_k = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_k)$. The n rows of $\mathbf{Y}_k$ are termed the principal coordinates in k dimensions (Gower, 1966). Hence, the classical MDS solution is to choose the configuration in $\mathbb{R}^k$ whose coordinates are determined by the first k eigenvectors of $\bar{\mathbf{T}}$. Such geometrical representation of proximity data is termed ordination.

The classical scaling method can also be applied to the matrix of similarities $S = (s_{uv})$ where $(0 \leq s_{uv} \leq 1)$ and $s_{uv} = s_{vu}$. To apply the above procedure, the similarities

can be converted into dissimilarities using some transformation, e.g. $d_{uv}^2 = (s_{uu} - 2s_{uv} + s_{vv})^{1/2}$, and $t_{uv} \equiv s_{uv}$ is used in matrix $\mathbf{T}$.

As given above, the starting point of principal coordinate analysis is an n-by-n matrix of (dis)similarities. If the procedure deals directly with the n-by-p original data matrix $\mathbf{X}$, the principal coordinate analysis is related to PCA (Borg and Groenen, 1997; Jolliffe, 2002). Suppose $\boldsymbol{\Sigma}$ denotes the sample covariance matrix. Then the (nonzero and distinct) eigenvalues of $\tilde{\mathbf{T}} = \mathbf{M}_n\mathbf{X}\mathbf{X}^\top\mathbf{M}_n$ are also the nonzero eigenvalues of $n\boldsymbol{\Sigma} = \mathbf{X}^\top\mathbf{M}_n\mathbf{X}$. Mardia *et al.* (1982) show the duality between principal coordinate analysis and PCA, and state that the principal coordinates of $\mathbf{X}$ in k dimensions are given by the centered scores of the n objects on the first k PCs (Mardia *et al.*, 1982, pp. 405).

2.7 Biplots

The concept of classical biplots (also called principal component biplots (Jolliffe, 2002)) was first developed and popularized by Gabriel (1971), but Gower and Hand (1996) reviewed much subsequent literature and considerably extended the idea. Consider an n-by-p centered data matrix $\mathbf{X}$ of rank r. Biplots provide plots of the n observations, together with the relative positions of the p variables, in fewer than r dimensions. The biplots construction begins with finding n row vectors $\mathbf{g}_i^\top$ and p row vectors $\mathbf{h}_j^\top$ such that each element in $\mathbf{X}$ is represented by their inner product. That is, if x_{ij} is the element in the ith row and the jth column of $\mathbf{X}$, then

$$x_{ij} = \mathbf{g}_i^\top\mathbf{h}_j, \quad i = 1, \ldots, n; \quad j = 1, \ldots, p. \tag{2.10}$$

The SVD of $\mathbf{X}$ is helpful in deriving biplots. Suppose

$$\mathbf{X} = \mathbf{ULA}^\top \tag{2.11}$$

where $\mathbf{U}$ is an n-by-r matrix of rank r with orthonormal columns $\mathbf{u}_i$, $\mathbf{A}$ is a p-by-r matrix of rank r with orthonormal vectors $\mathbf{a}_j$, and $\mathbf{L}$ is an r-by-r diagonal matrix of elements $\ell_1 \geq \ell_2 \geq \ldots \geq \ell_r > 0$. Define $\mathbf{L}^\beta$, a diagonal matrix with elements ℓ_j^β for $0 \leq \beta \leq 1$ $(j = 1, 2, \ldots, r)$. Then, $\mathbf{X}$ can be factorized into an n-by-r matrix $\mathbf{G}$ and a p-by-r matrix $\mathbf{H}$ as:

$$\mathbf{X} = \mathbf{ULA}^\top = \mathbf{UL}^\beta\mathbf{L}^{1-\beta}\mathbf{A}^\top = \mathbf{GH}^\top \tag{2.12}$$

where $\mathbf{G} = \mathbf{UL}^\beta$ and $\mathbf{H}^\top = \mathbf{L}^{1-\beta}\mathbf{A}^\top$. Thus, the vectors $\mathbf{g}_i^\top$ and $\mathbf{h}_j^\top$ are the rows of $\mathbf{G}$ and $\mathbf{H}$, respectively, each with r elements. Both $\mathbf{G}$ and $\mathbf{H}$ (called factors) are of rank r. The factorization can be made unique by orthogonal transformation (rotation or reflection), or by imposing a particular metric on the columns of $\mathbf{G}$ and $\mathbf{H}$.

In a matrix of rank 2, $\mathbf{g}_i$ and $\mathbf{h}_j$ are vectors of length two. Gabriel (1971) represented the np elements of $\mathbf{X}$ by the plots of the $n + p$ vectors $\mathbf{g}_i$ and $\mathbf{h}_j$, and called the plot a 'biplot' to stress the joint display of the row and column effect vectors in a r-dimensional space.

For any higher-rank matrix (that is $r > 2$), approximate biplots can be obtained after approximating the matrix by a rank-2 matrix. If $\mathbf{X}_{(2)}$ denotes a rank-2 approximation to the data matrix $\mathbf{X}$, then

$$\mathbf{X} = \mathbf{GH}^\top \approx \mathbf{G}_{(2)}\mathbf{H}_{(2)}^\top = \mathbf{X}_{(2)}, \tag{2.13}$$

where $\mathbf{G}_{(2)} = (\mathbf{u}_1, \mathbf{u}_2)$ and $\mathbf{H}_{(2)} = (\ell_1\mathbf{a}_1, \ell_2\mathbf{a}_2)$. Alternatively, this can be given as $\mathbf{G}_{(2)} = (\ell_1\mathbf{u}_1, \ell_2\mathbf{u}_2)$ and $\mathbf{H}_{(2)} = (\mathbf{a}_1, \mathbf{a}_2)$. In the latter case, $\mathbf{G}_{(2)}$ gives the values of

the first two principal components and $\mathbf{H}_{(2)}$ gives the coefficients that determine the PCs (the eigenvectors). The idea can be extended to the case where the matrix is approximated by rank-k ($k < r$), in which

$$\mathbf{X} \approx \mathbf{G}_{(k)}\mathbf{H}_{(k)}^{\top}, \tag{2.14}$$

where $\mathbf{G}_{(k)}$ and $\mathbf{H}_{(k)}$ contain the first k columns of $\mathbf{G}$ and $\mathbf{H}$, respectively. But, for $k > 2$, the graphical representation is less clear.

Chapter 3

Interpretable dimension reduction

Chapter 2 gave a brief overview of the commonly used techniques for reducing the dimension of a multivariate data set. Most of the techniques produce solutions in the form of linear combinations of the original variables. The most popular and efficient method of this type is PCA.

Principal components (PCs) are really useful if they can be easily interpreted. However, each PC is a weighted sum of all the original variables, which can make their interpretation difficult, ambiguous and/or even impossible. The process of interpreting components in a multidimensional space is sometimes referred to as reification (Krzanowski, 1988).

Traditionally, PCs are considered easily interpretable if there are plenty of small component loadings indicating the negligible importance of the corresponding original variables. Thus, the 'classical' way for PCs simple interpretation is to ignore loadings whose absolute values are below some specified threshold. But, as Cadima and Jolliffe (1995) argue, ignoring small-magnitude loadings in the interpretation of PCs can be misleading, especially for PCs computed from a covariance matrix. Chipman and Gu

(2005) define an interpretable component as one having many of its coefficients zero (and hence forming a sparse component) or taking only a few distinct values. In fact, as Cadima and Jolliffe (2001) describe, it is difficult to envisage criteria that explicitly define interpretability.

In this chapter, we give a literature review on some of the approaches proposed towards interpretable dimension reduction, especially in relation to PCs. The approaches are classified and presented in three main categories – rotation, constrained optimization and subset selection. It might be important to note here that only PCs have the property of orthogonality and uncorrelatedness. Any other alternatives or approximations to PCs do not retain either one or both of these properties.

3.1 Rotation

Simple structure rotation (Jolliffe, 2002) is historically the first method to aid the interpretation of PCs. It is simply a change of the coordinate axes according to a certain simplicity criterion. Suppose that the number of PCs to retain and rotate is decided to be k. Rotating a p-by-k loading matrix $\mathbf{A}$ of the first k PCs is made by post-multiplying it by an orthogonal rotation matrix $\mathbf{Q}$:

$$\mathbf{B} = \mathbf{AQ}.$$

Then, $\mathbf{B}$ gives the matrix of rotated loadings. The rotation problem is to find $\mathbf{Q}$ based on some rotation criterion.

The most popular rotation criteria are varimax and quartimax. These are designed to drive the loadings of a component towards 0 or towards the maximum possible absolute value (which is 1 for normalized loadings). For the commonly used varimax

rotation (Kaiser, 1958), $\mathbf{Q}$ is chosen to maximize

$$f(\mathbf{B}) = \sum_{l=1}^{k} \left[\sum_{j=1}^{p} b_{jl}^4 - \frac{1}{p} \left(\sum_{j=1}^{p} b_{jl}^2 \right)^2 \right] \tag{3.1}$$

where b_{jl} is the (j, l)th element of $\mathbf{B}$. The varimax criterion was initially introduced in factor analysis (Bernaards and Jennrich, 2005; Browne, 2001; Kaiser, 1958), and later adapted to PCA (Jolliffe, 2002). In PCA, if normalized loadings are rotated, then $\sum_{j=1}^{p} b_{jl}^2 = 1$, and the criterion (3.1) reduces to

$$f(\mathbf{B}) = \sum_{l=1}^{k} \sum_{j=1}^{p} b_{jl}^4 - \frac{k}{p}. \tag{3.2}$$

There are, however, some drawbacks in the rotation approach to PCs (Jolliffe and Uddin, 2000; Jolliffe *et al.*, 2003). The main drawback is that the rotated loadings are usually still difficult to interpret. In addition, either the orthogonality of the vectors of component loadings or the uncorrelatedness of the component scores are, inevitably, lost after rotation. Furthermore, different choices of normalization constraints result in different solutions. To this effect, Jolliffe (1995) discusses the effect of three different normalization constraints: $\mathbf{a}_i^\top \mathbf{a}_i = \lambda_i$ (the ith eigenvalue), $\mathbf{a}_i^\top \mathbf{a}_i = 1$ and $\mathbf{a}_i^\top \mathbf{a}_i = \lambda_i^{-1}$, where $\mathbf{a}_i$ denotes the ith column of $\mathbf{A}$. The first normalization constraint results in non-orthogonal rotated loadings and correlated rotated components. The second constraint results in orthogonal rotated loadings but correlated components, while the third constraint results in uncorrelated rotated components but non-orthogonal loadings.

3.2 Constrained methods

As a result of the drawbacks of the rotation approach, many constrained methods have been proposed for producing simple PCs. Some of these are briefly outlined below.

3.2.1 Restricting the values of loadings

Hausman (1982) proposes a simplified version of PCs in which the weights (or loadings) can take values from some small set, like $\{-1, 0, 1\}$ or $\{1, 0\}$. For this purpose, he used an optimisation technique called the *branch-and-bound algorithm*, which works as described below (Hand, 1981).

Suppose a problem has a large set, say S, of possible solutions and the aim is to find $x \in S$ which optimizes a criterion $J = J(x)$. Assume we wish to maximize J. Choose (at random) an element y to provide an initial upper bound. Suppose that we are examining a subset S_i with upper bound of J denoted as J_i which is greater than the current maximum $J(y)$ (so that we can not reject S_i). Then the branch-and-bound algorithm works as follows:

1. Split S into $S_1, \ldots, S_q$.

2. Set $i = 1$.

3. Find an upper bound J_i on S_i or if S_i is a single element, z, evaluate it to give $J_i = J(z)$.

4. If $J_i < J(y)$, go to (5); otherwise, go to (7).

5. We can reject S_i. If S_i is the last subset of S go to (6); otherwise, set i to $i + 1$ and go to (3).

6. Now all subsets of S have been evaluated or rejected (i.e., either $x \in S$ which maximizes $J(x)$ has been found or there is no element of $x \in S$ such that $J(x) > J(y)$).

7. We can not reject S_i. If S_i is a single element go to (8); otherwise, go to (9).

8. Now the single element $S_i = z$ is a better solution than y, so replace $J(y)$ by $J(z)$. If S_i is the last subset of S go to (6); otherwise, set i to $i + 1$ and go to (3).

9. S_i is not a single element so we must consider its elements, again by branching and bounding. So, set $S \leftarrow S_i$ and go to (1).

The branch-and-bound algorithm searches for a single-element solution $x \in S$ that optimizes the criterion $J(x)$ by partitioning S into different subsets, S_i. It starts by selecting an element y at random so that $J(x)$ is compared to $J(y)$. The algorithm repeats until a solution with an optimal value has been found or none of the elements give better result than a random value. Step 9 shows that the whole algorithm should repeat when the current solution still contains more than one element. Step 6 gives the conditions under which the whole algorithm comes to an end.

Let $\boldsymbol{\nu}$ denotes a p-dimensional vector of parameters. Hausman (1982) defines the vector $\boldsymbol{\nu}$ as

$$\boldsymbol{\nu} = \alpha \mathbf{t}$$

for some real number α and some vector $\mathbf{t} = (t_1, \ldots, t_p)$ with elements t_j all chosen from $S = \{-1, 0, 1\}$. If each element of $\mathbf{t}$ is a member of S, then we say that $\mathbf{t} \in S^{(p)}$. For a given data matrix $\mathbf{X}$, let $f(\mathbf{X}; \boldsymbol{\nu})$ be the objective function and let $g(\mathbf{X}; \boldsymbol{\nu}) = \mathbf{0}$ give some constraints on $\boldsymbol{\nu}$. Then the interest is to solve the optimization problem

$$\max_{\mathbf{t},\alpha} f(\mathbf{X}; \alpha\mathbf{t}) \quad \text{such that} \quad g(\mathbf{X}; \alpha\mathbf{t}) = \mathbf{0} \quad \text{and} \quad \mathbf{t} \in S^{(p)}. \tag{3.3}$$

Hausman (1982) indicated that the above constrained approach can be applied to such statistical analyses as PCA, LDA, canonical correlation analysis and multiple

regression. In the case of PCA, it maximizes the variance and the first constrained principal component (CPC) is $\tilde{y}_1 = \tilde{\mathbf{a}}_1^\top \mathbf{x}$, subject to the constraint that $\tilde{\mathbf{a}}_1 = \alpha \mathbf{t}$ for some $\mathbf{t} \in S^{(p)}$. A branch-and-bound algorithm is used to determine $\tilde{\mathbf{a}}_1$. The second CPC is given by $\tilde{y}_2 = \tilde{\mathbf{a}}_2^\top \mathbf{x}$, but unlike the unconstrained PCA, $\tilde{\mathbf{a}}_1$ is not necessarily orthogonal to $\tilde{\mathbf{a}}_2$. The vector $\tilde{\mathbf{a}}_2$ may be found by substituting the partial correlation matrix of $\mathbf{x}$ given $\tilde{y}_1$ for $\mathbf{R}$ and then repeating the procedure used to find the first CPC. This may avoid the two vectors being equal.

Vines (2000) proposed an iterative algorithm called a '*simplicity preserving*' transformation that produces simple components from a variance-covariance matrix as an approximation to the PCs, where the coefficients are restricted to integers. The algorithm starts with a pair of orthogonal directions, say $\mathbf{d}_1$ and $\mathbf{d}_2$, in a p-dimensional space, and searches for a linear transformation that preserves the orthogonality of the directions

$$(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2) = (\mathbf{d}_1, \mathbf{d}_2)\mathcal{P} \qquad \text{with} \qquad \mathcal{P} = \begin{pmatrix} 1 & \iota_2^2 \beta \\ \beta & -\iota_1^2 \end{pmatrix},$$

where $\iota_1^2 = \mathbf{d}_1^\top \mathbf{d}_1$, and $\iota_2^2 = \mathbf{d}_2^\top \mathbf{d}_2$. If $\boldsymbol{\Sigma}$ is the variance-covariance matrix of the data with respect to the original axes $\mathbf{d}_1$ and $\mathbf{d}_2$, then the variance-covariance matrix of the data with respect to the new axes $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ is $\boldsymbol{\Sigma}^* = \mathcal{P}^\top \boldsymbol{\Sigma} \mathcal{P}$. In addition, if $\mathbf{d}_1$ and $\mathbf{d}_2$ are vectors of integers, then $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_2$ will also be vectors of integers provided that the values of β are restricted to $\beta = i/2^q$ or $\beta = 2^q/i$, $i = -2^q, -2^q + 1, \ldots, 2^q$. This results in

$$\left.\begin{aligned} \boldsymbol{\nu}_1 &= 2^q \mathbf{d}_1 + 2^q \beta \mathbf{d}_2 \\ \boldsymbol{\nu}_2 &= 2^q \beta \iota_2^2 \mathbf{d}_1 - 2^q \iota_1^2 \mathbf{d}_2 \end{aligned}\right\} \quad |\beta| \leq 1, \qquad \left.\begin{aligned} \boldsymbol{\nu}_1 &= 2^q \mathbf{d}_1/\beta + 2^q \mathbf{d}_2 \\ \boldsymbol{\nu}_2 &= 2^q \iota_2^2 \mathbf{d}_1 - 2^q \iota_1^2 \mathbf{d}_2/\beta \end{aligned}\right\} \quad |\beta| > 1.$$

Vines (2000) generalized the above simplicity preserving transformation to p orthog-

onal simple directions based on Jacobi's method. Based on simple examples, $q = 0$ is found to give good results. The usual case is that the first few resulting components (those with higher variances) are simpler than the later components, with respect to the magnitudes of the integer loadings.

An alternative simple component analysis is proposed by Rousson and Gasser (2004), who partition components into two as *block* (those having the same sign for all non-zero loadings) and *difference* (those having some strictly positive and some strictly negative loadings). One motivation to their simple component approach is that when a correlation matrix has an approximate block structure with b blocks, then b of the principal components might be replaced by b block components. They used an explicit definition of simplicity rather than optimizing a criterion of simplicity to obtain a simple loading structure. For this purpose, they set some conditions to be satisfied in relation to the b block and the $k - b$ difference components.

In addition, Rousson and Gasser (2004) defined different optimality criteria in relation to the percentage of variance explained by components. If $\mathbf{R}$ denotes a correlation matrix of a p-vector of random variables $\mathbf{x}$ and $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k)$ is a $p \times k$ matrix of loadings, then the optimality criterion recommended by Rousson and Gasser (2004) is given by

$$\text{Opt}(\mathbf{V}) = \frac{\text{trace}(\mathbf{V}^\top \mathbf{R}\mathbf{V}) - \sum_{i=2}^{k} \mathbf{v}_i^\top \mathbf{R}\mathbf{v}_{(i-1)} (\mathbf{v}_{(i-1)}^\top \mathbf{R}\mathbf{v}_{(i-1)})^{-1} \mathbf{v}_{(i-1)}^\top \mathbf{R}\mathbf{v}_i}{\text{trace}(\mathbf{\Lambda}_k)}$$

where $\mathbf{\Lambda}_k$ denotes a diagonal matrix containing the first k eigenvalues of $\mathbf{R}$.

Rousson and Gasser (2004) described a two-stage algorithm for obtaining the k simple components. On the first stage, the p variables are classified into b disjoint blocks, each corresponding to an approximate block structure in the correlation matrix.

An agglomerative hierarchical procedure is used to cluster the blocks. On the second stage, the simple difference components are defined. For this purpose, they use the fact from PCA that the jth eigenvector of $\mathbf{R}$ is equal to the first eigenvector of the matrix

$$\mathbf{R} - \mathbf{R}\mathbf{A}_{j-1}(\mathbf{A}_{j-1}^{\top}\mathbf{R}\mathbf{A}_{j-1})^{-1}\mathbf{A}_{j-1}^{\top}\mathbf{R}.$$

To make the simple components close to PCA, the jth simple component (or the $(j-b)$th simple difference component) is obtained by regressing the original variables on the first $j-1$ simple components and computing the first principal component of these residual variables.

Chipman and Gu (2005) introduced three classes of constraints on the coefficients of a PC for the sake of interpretability: homogeneity, contrast and sparsity. Homogeneity refers to the case where the coefficients are constrained to take only three distinct values, 0 or $\pm c$, for the ith direction $\boldsymbol{\nu}_i$ such that $\boldsymbol{\nu}_i^{\top}\boldsymbol{\nu}_i = 1$. Among all possible $\boldsymbol{\nu}_i$, the best one can be obtained by either minimizing the angle to the ith PC direction, $\arccos(\mathbf{a}_i^{\top}\boldsymbol{\nu}_i)$, or (equivalently) maximizing the inner product $\mathbf{a}_i^{\top}\boldsymbol{\nu}_i$ over $\{-c, 0, c\}$ values, where $\mathbf{a}_i$ represents the ith PC direction. The search algorithm works in the following way: among all possible $\boldsymbol{\nu}_i$ with m non-zero elements, identify the m elements of $\mathbf{a}_i$ with the largest absolute values and set the corresponding elements of $\boldsymbol{\nu}_i$ to $\pm 1/\sqrt{m}$, matching signs with that of $\mathbf{a}_i$. All other elements of $\boldsymbol{\nu}_i$ are set to 0 with $\boldsymbol{\nu}_i^{\top}\boldsymbol{\nu}_i = 1$. Repeat this procedure for $m = 1, 2, \ldots, p$. Then, the $\boldsymbol{\nu}_i$ closest to $\mathbf{a}_i$ is identified. Similarly, the contrast constraint refers to the case where the coefficients of the ith direction $\boldsymbol{\nu}_i$ take the values $-c_1$, 0, and c_2 such that $\boldsymbol{\nu}_i^{\top}\mathbf{1}_p = 0$ and $\boldsymbol{\nu}_i^{\top}\boldsymbol{\nu}_i = 1$. The sparsity constraint is an attempt to set as many coefficients to zero as possible. It was approached by minimizing the angle (θ) between the sparse component ($\boldsymbol{\nu}_i$) and

its corresponding principal component directions ($\mathbf{a}_i$). Since the angle is minimized when $\boldsymbol{\nu}_i \equiv \mathbf{a}_i$, a criterion

$$C1 = \theta/(\pi/2) + \eta m/p$$

is introduced to be minimized over $\boldsymbol{\nu}_i$ and m, where m (the number of nonzero coefficients) is added as a penalty term, and η is a tuning parameter.

The idea of Chipman and Gu (2005) is further studied and elaborated by Anaya-Izquierdo *et al.* (2010). The approach is developed in such a way that each eigen-vector is replaced by a simple vector, close to it in angle terms, whose entries are small integers while preserving orthogonality. It is an exploratory approach, where a range of sets of pairwise orthogonal simple components are systematically obtained, from which the user may choose.

3.2.2 The simplified component technique

Jolliffe and Uddin (2000) propose a method called the *simplified component technique* (SCoT) as an alternative to rotation techniques in PCA . If $f(\boldsymbol{\nu}_l)$ denotes the varimax simplicity criterion given by (3.1) for a single factor $\boldsymbol{\nu}_l$, and $V(\boldsymbol{\nu}_l)$ is the variance of the lth simple component $\boldsymbol{\nu}_l^\top \mathbf{x}$, then the SCoT successively maximizes

$$V(\boldsymbol{\nu}_l) + \xi f(\boldsymbol{\nu}_l) \tag{3.4}$$

subject to $\boldsymbol{\nu}_l^\top \boldsymbol{\nu}_l = 1$, and (for $l \geq 2$), $\boldsymbol{\nu}_i^\top \boldsymbol{\nu}_l = 0$, $i < l$, where ξ is a simplicity/complexity parameter.

3.2.3 A modified principal component technique based on the LASSO

Jolliffe *et al.* (2003) develop a modified PC, called the SCoTLASS, based on the LASSO (Tibshirani, 1996). The SCoTLASS introduces extra constraints

$$\sum_{j=1}^{p} |a_{ij}| \leq t \ , \ \text{for } i = 1, 2, \ldots, p, \tag{3.5}$$

to the standard PCA for some tuning parameter t, where a_{ij} is the jth element of the ith vector of component loadings. It is indicated that for $t < \sqrt{p}$, decreasing the value of t progressively decreases the number of variables with nonzero loadings. On the other hand, $t \geq \sqrt{p}$ gives PCA, and $t = 1$ leads to the case where only one variable gets nonzero-loading. SCoTLASS solves a non-convex constrained optimization problem, and is computationally expensive. Witten *et al.* (2009) propose a new algorithm for solving the SCoTLASS problem.

A complementary approach to the numerical solution of the SCoTLASS was also considered by Trendafilov and Jolliffe (2006) based on the projected gradient approach by introducing an exterior penalty function. It is a method based on the classical gradient approach and modified for analyzing and solving constrained optimization problems.

Trendafilov and Jolliffe (2007) use a similar idea for simplifying the interpretation of Fisher discriminant function coefficients. They imposed the LASSO constraint on the standard linear discriminant analysis (LDA). Considering the LDA problem outlined in Section 2.3, additional constraints

$$\|\boldsymbol{\omega}_i\| \leq t_i, \ i = 1, \ldots, k,$$

are imposed on the vector of loadings with $t_i \in [1, \sqrt{p}]$, which drive many coefficients to be exactly zero based on the magnitude of t_i, where $\|.\|$ denotes the L_1 norm.

3.2.4 Sparse principal components

Zou *et al.* (2006) introduced a modified PCA method called sparse principal component analysis (SPCA). They first transform the PCA problem to a regression-type problem to derive PCs. The idea behind this approach is that, as each PC is a linear combination of all the p variables, its loadings can be recovered by regressing the PC on the p variables. Consider $\tau > 0$ and the SVD

$$\mathbf{X} = \mathbf{ULA}^\top$$

with the scores of ith PC $\mathbf{y}_i = \ell_i \mathbf{u}_i$, where $\mathbf{u}_i$ is the ith column of $\mathbf{U}$ and ℓ_i is the (i,i)th element of $\mathbf{L}$. From the ridge regression estimates

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}} \left(||\mathbf{y}_i - \mathbf{Xv}||^2 + \tau ||\mathbf{v}||^2 \right), \tag{3.6}$$

let $\hat{\boldsymbol{\beta}}_i = \hat{\mathbf{v}}/||\hat{\mathbf{v}}||$. Then $\hat{\boldsymbol{\beta}}_i = \mathbf{a}_i$, the loadings of the ith PC (Zou *et al.*, 2006).

Now, let $\mathbf{x}_i$ denotes the ith row of $\mathbf{X}$, and

$$(\hat{\mathbf{z}}, \hat{\mathbf{v}}) = \arg\min_{\mathbf{z},\mathbf{v}} \sum_{i=1}^{n} \left(||\mathbf{x}_i - \mathbf{z}\mathbf{v}^\top \mathbf{x}_i||^2 + \tau ||\mathbf{v}||^2 \right) \tag{3.7}$$

subject to $||\mathbf{z}||^2 = 1$. Then, $\hat{\mathbf{v}}$ is proportional to $\mathbf{a}_1$. If the first k PCs are considered with $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k]$, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k]$, and

$$(\hat{\mathbf{Z}}, \hat{\mathbf{V}}) = \arg\min_{\mathbf{Z},\mathbf{V}} \sum_{i=1}^{n} \left(||\mathbf{x}_i - \mathbf{ZV}^\top \mathbf{x}_i||^2 + \tau \sum_{j=1}^{k} ||\mathbf{v}_j||^2 \right) \tag{3.8}$$

subject to $\mathbf{Z}^\top \mathbf{Z} = \mathbf{I}_k$, then $\hat{\mathbf{v}}_j$ is proportional to $\mathbf{a}_j$, $j = 1, 2, \ldots, k$.

The LASSO approach is used to produce sparse loadings, by adding the LASSO penalty $\sum_{j=1}^{k} \tau_{1,j} ||\mathbf{v}_j||_1$ to the criterion in (3.8). Different $\tau_{1,j}$'s are used for penalizing

the loadings of different principal components. Li (2007) followed the idea of Zou *et al.* (2006) to get sparse sufficient dimension reduction for LDA.

Unlike ordinary PCs, sparse PCs are in general correlated with each other, and thus, the sum of their variances might not show the real explained variance. Actually, only PCs satisfy the properties of orthogonality and uncorrelatedness simultaneously. All alternatives to the ordinary PCs sacrifice either one or both of these properties. As a result, Zou *et al.* (2006) propose a method for computing variances of components adjusted for their correlation. Let $\mathbf{V}_k$ denotes the matrix of sparse loadings for the first k sparse PCs of $\boldsymbol{\Sigma}$. The diagonal elements of the matrix $\mathbf{S}_k = \mathbf{V}_k^\top \boldsymbol{\Sigma} \mathbf{V}_k$ give the vector of (unadjusted) variances of the sparse components. If $\mathbf{F}_k$ denotes the upper triangular matrix of the Cholesky factorization of $\mathbf{S}_k$, then the vector of adjusted variances is given by the squared diagonal elements of $\mathbf{F}_k$. Obviously, if $\mathbf{V}_k$ is the matrix of PC loadings, the adjusted variances are the same as the variances of the PCs.

Gervini and Rousson (2004) are also concerned with the evaluation of correlated components. They argue that a criterion for evaluating dimension-reducing components should satisfy at least two conditions: generality and uniqueness. Generality refers to the applicability of the criteria to a wide range of components, while uniqueness limits the variance maximization criteria only to the PCs. They proposed two additional criteria to satisfy the condition of uniqueness. The first criterion, which is related to the sum of variances corrected for correlation, states "if a new component $y_q = \mathbf{a}_q^\top \mathbf{x}$ is added to a system of $q-1$ components, an indicator of the *real* contribution of y_q to the total variance of the system is the residual variance of the linear prediction of y_q given the first $q-1$ components" (Gervini and Rousson, 2004, pp.

75). For a $p \times k$ ($k \leq p$) loading matrix $\mathbf{A}$ and a p-vector $\mathbf{x}$ with covariance matrix $\boldsymbol{\Sigma}$, they propose a 'corrected sum of variances' (CSV) criterion given by

$$\mathrm{CSV}(\mathbf{A}) = \frac{\sum_{q=1}^{k} \left(\mathbf{a}_q^\top \boldsymbol{\Sigma} \mathbf{a}_q - \mathbf{a}_q^\top \boldsymbol{\Sigma} \mathbf{A}_{(q-1)} \left(\mathbf{A}_{(q-1)}^\top \boldsymbol{\Sigma} \mathbf{A}_{(q-1)} \right)^{-1} \mathbf{A}_{(q-1)}^\top \boldsymbol{\Sigma} \mathbf{a}_q \right)}{\sum_{q=1}^{k} \lambda_q} \tag{3.9}$$

where λ_q is the qth eigenvalue of $\boldsymbol{\Sigma}$ and $\mathbf{A}_{(q)} = (\mathbf{a}_1, \ldots, \mathbf{a}_q)$. The criterion is said to satisfy the conditions of generality and uniqueness. Due to the invariant property of CSV under permutation of components, they propose a second criterion, called 'symmetrically corrected sum of variances', which is given by replacing all the $\mathbf{A}_{(q-1)}$ terms in (3.9) by $\mathbf{A}_{-q}$, a $p \times (k-1)$ matrix obtained after deleting the qth column of $\mathbf{A}$.

Another kind of sparse PCA is introduced by d'Aspremont *et al.* (2007) as a cardinality-constrained quadratic program. For a given covariance matrix $\boldsymbol{\Sigma}$, the quadratic form $\mathbf{v}^\top \boldsymbol{\Sigma} \mathbf{v}$ is maximized subject to $\mathbf{v}$ having no more than m non-zero elements, i.e.

$$\max_{\substack{\mathbf{v}^\top \mathbf{v} = 1 \\ \mathrm{card}(\mathbf{v}) \leq m}} \mathbf{v}^\top \boldsymbol{\Sigma} \mathbf{v} \,, \tag{3.10}$$

where cardinality of a vector refers to the number of its nonzero elements. The sparseness is controlled by the value of m. The quadratic optimization subject to cardinality constraint is hard to solve, but d'Aspremont *et al.* (2007) relaxed it to a semidefinite program. Despite the advanced numerical technique, the choice of the cardinality presents very much the same problem as with the LASSO threshold in SCoTLASS and SPCA. Probably the most elegant approach that swiftly treats both LASSO and cardinality constraints was recently proposed by Journée *et al.* (2008). Nevertheless, the LASSO/cardinality related approaches to sparseness are numerically demanding

while leaving freedom for subjective interpretation. Necessarily, they are followed by some kind of validation of the threshold/cardinality, which may not be feasible for large data sets.

Moghaddam *et al.* (2006) proposed an algorithm for sparse PCA problem (3.10) based on the *inclusion principle* for eigenvalue bounds. Let $\mathbf{\Sigma}_k$ be the $k \times k$ principal submatrix of $\mathbf{\Sigma}$ with eigenvalues $\lambda_i(\mathbf{\Sigma}_k)$. Then, for every integer i, $1 \leq i \leq k$,

$$\lambda_i(\mathbf{\Sigma}) \leq \lambda_i(\mathbf{\Sigma}_k) \leq \lambda_{i+p-k}(\mathbf{\Sigma}) \tag{3.11}$$

holds for $1 \leq k \leq p$. The best strategy for the algorithm is found to be a bi-directional greedy search, called greedy sparse PCA (GSPCA). [Greedy algorithm, as defined by National Institute of Standards and Technology (`http://xw2k.nist.gov/dads/html/greedyalgo.html`), is an algorithm that always takes the best immediate, or local, solution while finding an answer. The word 'greedy' is used from the fact that such algorithm examines each entity at most once and decides its fate once and for all during that examination.] They also defined an algorithm called exact sparse PCA (ESPCA) which is guaranteed to terminate with the optimal solution. As a cost-effective strategy, they recommend using both methods simultaneously.

Johnstone and Lu (2004) considered sparse PCA for a dataset $\{\mathbf{x}_i \in \mathbb{R}^p,\ i = 1, \ldots, n\}$ in which the number of variables p is comparable to the number of observations n, or may even be larger (example, high-dimensional signals or images). For such data sets, they propose some initial reduction in dimensionality before applying any PCA-type search, which can best be achieved by working in a basis in which the signals have a sparse representation.

3.3 Subset selection

3.3.1 Selecting subsets of variables

McCabe (1984) proposes using an optimality criteria for selecting a subset of variables (called *principal variables*) that contain as much information as possible. Assume that $\mathbf{x}$ is a p-dimensional normally distributed random vector with mean zero and known positive definite covariance matrix $\mathbf{\Sigma}$. Consider all possible partitions $\mathbf{x}_1$ and $\mathbf{x}_2$ of $\mathbf{x}$, where $\mathbf{x}_1$ is the vector of k retained variables and $\mathbf{x}_2$ is the $(p-k)$-vector of discarded variables. Up to a row-and-column permutation, the corresponding partition of $\mathbf{\Sigma}$ holds

$$\begin{pmatrix} \mathbf{\Sigma}_{11} & \mathbf{\Sigma}_{12} \\ \mathbf{\Sigma}_{21} & \mathbf{\Sigma}_{22} \end{pmatrix},$$

where $\mathbf{\Sigma}_{11}$ is the $k \times k$ covariance matrix of $\mathbf{x}_1$. Then, selection of a set of k variables is equivalent to selection of $k \times k$ matrix $\mathbf{\Sigma}_{11}$ from all possible choices. The optimality criteria for PCs and other related criteria considered in McCabe (1984) were then applied for the optimal choice of $\mathbf{\Sigma}_{11}$.

Cadima and Jolliffe (2001) considered the problem of identifying subsets of variables which best approximate the full set of variables or their first few PCs. They stress dimensionality reduction in terms of the original variables, rather than derived variables (or PCs) whose definition requires all the original variables. Consider an $n \times p$ data matrix $\mathbf{X}$ of rank p with sample covariance matrix $\mathbf{\Sigma}$. From the spectral decomposition $\mathbf{\Sigma} = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^\top$, the columns of the matrix $\mathbf{XA}$ give the PCs of the data. Let $\mathcal{T}$ represent the subspace spanned by any q PCs, let $\mathcal{K}$ represent the subspace spanned by any k of the original variables which are considered to approximate these

PCs and let the indices of these k variables be collected in a set of integers κ. Then the generalized coefficient of determination (GCD) is used as a criterion for similarity between the subspaces $\mathcal{T}$ and $\mathcal{K}$. The expression for the GCD is given as

$$\text{GCD}(\mathcal{T},\mathcal{K}) = \frac{1}{\sqrt{qk}} \sum_{i\in\kappa} \lambda_i \mathbf{a}_i^{\kappa\top} \boldsymbol{\Sigma}_\kappa^{-1} \mathbf{a}_i^\kappa = \frac{1}{\sqrt{qk}} \sum_{i\in\kappa} (\rho_m)_i^2, \tag{3.12}$$

where $\boldsymbol{\Sigma}_\kappa$ is the $k \times k$ submatrix of $\boldsymbol{\Sigma}$ that results from retaining the k rows/columns whose row/column numbers are in κ, $\mathbf{a}_i$ denotes the ith eigenvector of $\boldsymbol{\Sigma}$ (column of $\mathbf{A}$) with the corresponding eigenvalue λ_i and $\mathbf{a}_i^\kappa$ denotes the subvector of $\mathbf{a}_i$ that results from retaining only those elements in positions given by the set κ. The value $(\rho_m)_i$ is the multiple correlation between the ith PC and the k variables spanning $\mathcal{K}$. Cadima and Jolliffe (2001) suggest using stepwise algorithm to select a subset of variables, once a criterion is identified.

Wood *et al.* (2005) propose a method of variable selection in discriminant analysis. The objective is to find a subset of original features which can discriminate between the groups as successfully as possible compared to the full set of features. Given an $n \times p$ data matrix $\mathbf{X}$, they introduced a new $n \times g$ matrix $\mathcal{M}$ which defines the group structure of the data by taking a 1 in position (i, j) if the ith row of $\mathbf{X}$ belongs to group j. Denote by $\mathbf{1}_n$ a vector of n 1's, and define the projection matrix associated with $\mathcal{M}$ as $\mathbf{P}_\mathcal{M} = \mathcal{M}(\mathcal{M}^\top\mathcal{M})^{-1}\mathcal{M}^\top$ and the projection matrix associated with $\mathbf{1}_n$ as $\mathbf{P}_{\mathbf{1}_n} = \mathbf{1}_n(\mathbf{1}_n^\top\mathbf{1}_n)^{-1}\mathbf{1}_n^\top = \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$. Define $\mathbf{J} := (\mathbf{I}_n - \mathbf{P}_{\mathbf{1}_n})\mathbf{X}$ and consider the matrices of sums of squares and cross-products

$$\mathbf{T}_0 = \mathbf{J}^\top\mathbf{J} \tag{3.13}$$

and

$$\mathbf{B}_0 = \mathbf{J}^\top(\mathbf{P}_\mathcal{M} - \mathbf{P}_{\mathbf{1}_n})\mathbf{J}. \tag{3.14}$$

Let κ be the set of k integers chosen from the set $\{1,\ldots,p\}$ to identify the subset of variables, and $\mathbf{I}_\kappa$ be the $p\times k$ matrix formed from the $p\times p$ identity matrix by removing those columns not in κ. Denote $\mathbf{T}_\kappa = \mathbf{I}_\kappa^\top \mathbf{T}_0 \mathbf{I}_\kappa$, $\mathbf{B}_\kappa = \mathbf{I}_\kappa^\top \mathbf{B}_0 \mathbf{I}_\kappa$, and $t = \min\{k, g-1\}$. Then Wood *et al.* (2005) used a genetic algorithm to maximize the measure of Yanai's Generalized Coefficient of Determination (GCD), given by

$$\text{GCD} = \frac{\text{tr}[\mathbf{T}_\kappa^{-1}\mathbf{B}_\kappa]}{\sqrt{t(g-1)}}, \tag{3.15}$$

in order to find the best subsets of variables.

3.3.2 Feature selection and extraction

Pattern recognition usually deals with information processing problems such as speech recognition, classification of handwritten characters, and so on, each of which containing a large number of input variables (Webb, 1999).

One method to reduce the number of variables is to combine the input variables together to make a smaller number of new variables called features (Bishop, 1995). Patterns in a classical pattern recognition techniques are represented as a vector of feature values, and feature selection and extraction methods are important techniques for such problems.

Feature selection deals with choosing the 'best' possible subset of size k from a set of p features according to an objective function. An optimal search procedure is the branch-and-bound procedure (Section 3.2.1), a top-down procedure in which we start with the full set of p variables and construct a tree by deleting redundant variables successively. It is, however, computationally expensive for large p. There are several suboptimal search algorithms (Webb, 1999).

Feature extraction is a method in which a linear transformation of an $n \times k$ pattern matrix $\mathbf{Y}$ is derived from a given $n \times p$ pattern matrix $\mathbf{X}$, where $\mathbf{Y} = \mathbf{XA}$ and $\mathbf{A}$ is a $p \times k$ ($k < p$) transformation matrix (Raymer *et al.*, 2000). Criteria for feature extraction can be based on unsupervised setting (that aims to minimize the information loss, e.g. PCA), or supervised setting (that aims to maximize the class discrimination, e.g. LDA). For PCA, the columns of $\mathbf{A}$ consist of the eigenvectors of the covariance matrix of the given patterns.

Chapter 4

sBarse: sparse biplots component analysis

In this chapter, a very simple method for computing simple components (SCs) is proposed. Sparse biplots component analysis, or sBarse for short, proceeds as follows. Sparse loadings are constructed from the biplots of the input data, either the data matrix or the sample correlation matrix. The resulting sBarse components have orthogonal loadings, each original variable corresponding to only one sBarse component and, thus, leading to easily interpretable components. This contrasts with many existing methods producing SCs with non-orthogonal loadings and/or overlapping variables, for example Chipman and Gu (2005), d'Aspremont *et al.* (2007), Moghaddam *et al.* (2006) and Witten *et al.* (2009). The sparseness of the sBarse solution and the number k of the SCs involved are chosen to maximize the adjusted variance of the sBarse components and to be as close as possible to the input data in terms of the RV-coefficient (Robert and Escoufier, 1976).

The chapter is organized as follows. An intuitive introduction to the sBarse method

is given in Section 4.1, followed by a more formal treatment in Section 4.2. In Section 4.3, the sBarse method is tested, and compared with other similar methods, on the benchmark Jeffers's Pitprop data (Jeffers, 1967). A simulated data set is also used to test the performance of the method. Then, the sBarse method is applied to study a real gene expression data set concerning breast cancer, a case where the number of variables is far larger than the sample size. This data set is used by Chin *et al.* (2006) and is freely available from `http://icbp.lbl.gov/breastcancer/`. For comparison, we use the subset of the gene expression data set considered in Witten *et al.* (2009). A brief summary of the chapter is given in Section 4.4.

4.1 Sparse principal components

4.1.1 Rationale

There are a number of different ways to achieve PC simplification, as listed in Chapter 3. The proposed new method produces simplified loadings for all components simultaneously in contrast to most of the existing methods where each PC is simplified separately from the others.

Let $\mathbf{A}$ be a $p \times p$ orthogonal matrix of PC loadings, whose jth column represents the jth eigenvector of the correlation matrix $\mathbf{R}$ corresponding to the jth largest eigenvalue λ_j, $j = 1, 2, \ldots, p$. Then, each loading a_{ij} in $\mathbf{A}$ represents the contribution of the ith original variable x_i in the jth PC. The aim is to simplify the loadings by comparing the contribution of a variable to each of the PCs, so that the resulting SCs are easier to interpret. The idea is that a variable will be retained only in the SC in which it is most important. This can be easily achieved as follows.

Assume that each row of **A** represents a point in $\mathbb{R}^p$ and let $\mathbf{e}_q$ denote the qth coordinate vector of $\mathbb{R}^p$, i.e. $\mathbf{e}_q = [\underbrace{0,...,0}_{q-1}, 1, \underbrace{0,...0}_{p-q}]$. Consider approximation of the ith row of **A** by the nearest $\mathbf{e}_q$ or $-\mathbf{e}_q$, for $q = 1, 2, \ldots, p$. This requires finding the least Euclidean distance between the ith row of **A** and all possible $2p$ vectors $\mathbf{e}_q$ and $-\mathbf{e}_q$.

For instance, in a 2-dimensional space, a row in a 2×2 matrix of loadings **A** can be approximated by either of the following: $(1,0)$, $(0,1)$, $(-1,0)$, or $(0,-1)$. That is, after approximation, only one of the coefficients on the row of **A** take the value 1 (or -1 if the original loading is negative) and the remaining coefficients take 0.

The above approximation procedure uses only the unweighted loadings **A** and does not take into account the variances of the PCs. To take into account that the first few PCs explain the majority of the variation in the data, consider the following matrix of weighted loadings **B** with elements defined by $b_{ij} = \sqrt{\lambda_j} \times a_{ij}$. Since the eigenvalues are in decreasing order of magnitude, more weights are being given to the first few PCs.

Let δ_{qj} be the Kronecker delta for $j = 1, 2, \ldots, p$ and $q = 1, 2, \ldots, p$. The Euclidean squared distances, between the ith row of **B** and each of the $2p$ unit vectors $\mathbf{e}_q$ and $-\mathbf{e}_q$ are:

$$\sum_{j=1}^{p} (b_{ij} \pm \delta_{qj})^2, q = 1, 2, \ldots, p \; . \tag{4.1}$$

For the ith row of **B**, the minimal distance (4.1) is achieved for $\text{sgn}(b_{iq})\mathbf{e}_q$ for that value of q for which $|b_{iq}|$ (or b_{iq}^2) is maximal, $\text{sgn}(b_{iq})$ denoting the sign of b_{iq}. The vector $\mathbf{e}_q$ (or $-\mathbf{e}_q$) that gives the smallest value of (4.1) is the required approximation for the ith row of **B** and is collected as the ith row of a $p \times p$ matrix **V**. The same is repeated for all rows $i = 1, \ldots, p$. The resulting matrix **V** has exactly one 1 (or -1) in each row, but the number of 1s (or -1s) in a column may vary between 0 and p.

Note, that the total number of non-zero elements in $\mathbf{V}$ is p. Finally, the first k ($\leq p$) nonzero-columns of $\mathbf{V}$ are normalized into, say, $\tilde{\mathbf{V}}_k$ by dividing each column of $\mathbf{V}$ by the square-root of the number of non-zero elements in the column. The columns of $\tilde{\mathbf{V}}_k$ contain the loadings for the first k sparse components (SCs). This idea is extended to the sBarse method in Section 4.2.

The main aim of SCs is to simplify interpretation. However, there is one more advantage gained: it can help to find the appropriate number k of components to retain. Indeed, the weighted loadings in the ith row

$$\lambda_1 a_{i1}^2 \quad \lambda_2 a_{i2}^2 \quad \dots \quad \lambda_p a_{ip}^2, \quad i = 1, 2, \dots, p$$

are the values to be compared in the process of approximation. Since the eigenvalues are in decreasing order of magnitudes the last $p - k$ terms are systematically reduced. This implies that the approximating $\mathbf{e}_q$ (or $-\mathbf{e}_q$) will be most likely for some $q \in [1, k]$. In this case, the last $p-k$ columns in all rows will be identically zero and so the matrix of original loadings $\mathbf{A}$ can be approximated only by the first k orthogonal SC loadings, i.e. can be represented in a reduced k-dimensional subspace of $\mathbb{R}^p$. The estimation of k might be used as an alternative to the scree plot and the cumulative percentage of variance explained for deciding the number of PCs to retain (Jolliffe, 2002, p.115).

4.1.2 Correlation as a criterion

The approximation procedure outlined in Section 4.1.1 has the following meaning. Since the PCs are uncorrelated, the squared correlation between the ith variable and the jth PC is $\rho_{ij}^2 = \lambda_j a_{ij}^2$, where λ_j and a_{ij} are the variance and the ith loading of the jth PC (Jolliffe, 2002, p.25). This ρ_{ij}^2 is the same as the squared weighted loading,

b_{ij}^2, considered in Section 4.1.1. Hence, the procedure for making $\mathbf{B}$ sparse can be interpreted in terms of correlations as follows. Consider the ith row of the weighted matrix $\mathbf{B} = (b_{ij})$, where $b_{ij}^2 = \rho_{ij}^2$ gives the correlation of the ith variable and the jth PC for $j = 1, 2, \ldots, p$. Then, the aim is to relate the ith variable with the jth PC for which ρ_{ij}^2 is the largest. For the ith row of $\mathbf{B}$, replace by ± 1 the jth element for which ρ_{ij}^2 is the largest and by 0 all the others. Here, the ith variable is being related to a particular PC based on its explanatory power. The same is repeated for all rows of $\mathbf{B}$ until the sparse matrix $\mathbf{V}$ is obtained. As a result, the method only holds for PCA based on the correlation matrix and not for covariance-based PCA.

The approximation procedure considers only the component for which each particular variable is most important. A natural generalization is to introduce a tuning parameter measuring the variable importance and consider more than one component for which particular variable is relatively important. Such generalized approximation will be studied elsewhere.

The following example illustrates the approximation procedure using a well known data set.

Example 1: The Pitprop data contains 13 variables measured for 180 pitprops cut from Corsican pine timber (Jeffers, 1967). Denote by $x_1, x_2, \ldots, x_{13}$ the variables in the order they appear in the cited paper. Unfortunately, the raw Pitprop data seem lost, and only their correlation matrix is available. This data set is already a standard example in any work on sparse approximation of PCA. Note that the correlation coefficient $r_{5,11} = 0.091$ in (Jolliffe, 2002, Table 8.2) should be $r_{5,11} = -0.091$ (Jeffers, 1967, Table 2).

Jeffers (1967) and many other authors chose the first six PCs for further analysis.

Their loadings and the variances explained by them are given in the left hand side block of Table 4.1. Then, the procedure discussed in Section 4.1.1 is applied to the correlation matrix and the solution $\tilde{\mathbf{V}}_6$ is given in the right hand side block of Table 4.1. The last seven SCs are identically zero.

Thus, the interpretation of the components for the Pitprop data will be based on the first six SCs. The first SC is a weighted sum of the variables 1, 2, 7, 8, 9 and 10, and represents the overall size of the prop. The second sparse component, with nonzero weights for variables 3 and 4, measures the degree of seasoning. The third sparse component, with nonzero weights for variables 5 and 6, is a measure of the rate of growth of the timber. The fourth, fifth, and sixth sparse components are composed by single variables 11, 12 and 13, respectively.

Table 4.1: *Loadings and percentage of cumulative variance (%Cvar) & adjusted variance ($\%Cvar_{adj}$) of the first six PCs and the corresponding SCs, Jeffers's Pitprop data. Empty cells have zero values.*

	Principal Components						Sparse Components					
Variable	1	2	3	4	5	6	1	2	3	4	5	6
x_1	-.40	.22	-.21	-.09	-.08	.12	-.41					
x_2	-.41	.19	-.24	-.10	-.11	.16	-.41					
x_3	-.12	.54	.14	.08	.35	-.28		.71				
x_4	-.17	.46	.35	.05	.36	-.05		.71				
x_5	-.06	-.17	.48	.05	-.18	.63			.71			
x_6	-.28	-.01	.48	-.06	-.32	.05			.71			
x_7	-.40	.19	.25	-.07	-.22	.00	-.41					
x_8	-.29	-.19	-.24	.29	.19	-.06	-.41					
x_9	-.36	.02	-.21	.10	-.10	.03	-.41					
x_{10}	-.38	-.25	-.12	-.21	.16	-.17	-.41					
x_{11}	.01	.21	.07	.80	-.34	.18				1		
x_{12}	.12	.34	.09	-.30	-.60	-.17					-1	
x_{13}	.11	.31	-.33	-.30	-.08	.63						1
%Cvar	32.5	50.7	65.2	73.7	80.7	87.0	28.8	43.3	53.8	61.5	69.2	76.8
$\%Cvar_{adj}$	32.5	50.7	65.2	73.7	80.7	87.0	28.8	42.9	52.5	59.9	66.7	73.3

4.2 Computing sBarse components

The sBarse method was introduced intuitively in Section 4.1. Here, we consider the method in a more formal way. First, it is shown that the approximation procedure introduced in Section 4.1 is a special case of correlation biplot construction. Then, the general sBarse method is presented as a method for seeking the correlation biplot which maximizes a criterion involving the RV-coefficient (Robert and Escoufier, 1976)

and the variance explained. In this sense, the sBarse solution achieves an optimal fit to the data.

4.2.1 Biplots and their goodness-of-fit

We start with a brief summary of biplots (see Section 2.7) and of measures of their goodness-of-fit to the data (Gabriel, 1971; Gower and Hand, 1996).

Let $\mathbf{X}$ be a standardized $n \times p$ data matrix of rank r, with a SVD

$$\mathbf{X} = \mathbf{ULA}^\top, \tag{4.2}$$

where $\mathbf{U}$ and $\mathbf{A}$ are $n \times r$ and $p \times r$ orthonormal matrices, and $\mathbf{L}$ is $r \times r$ diagonal matrix of singular values $\ell_1 \geq \ell_2 \geq \cdots \geq \ell_r > 0$. Let $\mathbf{L}^\beta$ $(0 \leq \beta \leq 1)$ be the diagonal matrix whose elements are $\ell_1^\beta, \ell_2^\beta, \ldots, \ell_r^\beta$ so that (4.2) can be rewritten as:

$$\mathbf{X} = \mathbf{UL}^{1-\beta}\mathbf{L}^\beta\mathbf{A}^\top. \tag{4.3}$$

Let $\mathbf{U}_k = [\mathbf{u}_1, ..., \mathbf{u}_k]$, $\mathbf{A}_k = [\mathbf{a}_1, ..., \mathbf{a}_k]$ and $\mathbf{L}_k = \begin{pmatrix} \ell_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \ell_k \end{pmatrix}$ for any $k \in [1, r]$. Put $\mathbf{G}_{\beta,k} := \mathbf{U}_k\mathbf{L}_k^{1-\beta}$ and $\mathbf{H}_{\beta,k} := \mathbf{A}_k\mathbf{L}_k^\beta$. Then, the following rank k least-squares approximation holds:

$$\mathbf{X} = \mathbf{GH}^\top \approx \mathbf{G}_{\beta,k}\mathbf{H}_{\beta,k}^\top. \tag{4.4}$$

The matrices $\mathbf{G}_{\beta,k}$ and $\mathbf{H}_{\beta,k}$ are called biplot factors and their rows, called biplots, are the markers for the n rows (observations) and p columns (variables) of $\mathbf{X}$. Biplots are used to approximate the data $\mathbf{X}$ and can be constructed with any factors $\mathbf{G}_{\beta,k}$ and $\mathbf{H}_{\beta,k}$, and with any choice of $\beta \in [0, 1]$ and $k \leq r$. Interpretation of the most important biplots with $\beta = 0, \frac{1}{2}$ and 1 is given in Jolliffe (2002).

Biplots are also used to approximate the sample correlation matrix $\mathbf{R} = \mathbf{X}^\top\mathbf{X}$ (Gabriel, 1971; Gower and Hand, 1996, Ch 2, 11). We may call such biplots as correlation biplots, to differentiate them from those based on $\mathbf{X}$. These can also be constructed with any choice of $\beta \in [0, 1]$ as above, but with a single biplot factor $\mathbf{H}_{\beta,k} = \mathbf{A}_k\mathbf{L}_k^\beta$. For example, the choice of $k = 2$ and $\beta = 1$ gives the biplot factor $\mathbf{H}_2 = \mathbf{A}_2\mathbf{L}_2$, which gives the best two-dimensional least squares approximation of $\mathbf{R} \approx \mathbf{H}_2\mathbf{H}_2^\top$. This follows from the eigenvalue decomposition (EVD) of the sample correlation matrix $\mathbf{R} = \mathbf{A}\mathbf{L}^2\mathbf{A}^\top = \mathbf{A}\mathbf{\Lambda}\mathbf{A}^\top = \mathbf{H}\mathbf{H}^\top$, where $\mathbf{\Lambda} = \mathbf{L}^2$ contains the eigenvalues of $\mathbf{R}$. In general, consider the following biplot factor $\mathbf{B}_{\alpha,k} = \mathbf{A}_k\mathbf{\Lambda}_k^\alpha$ of rank k with $\alpha \in [0, 1]$. Then, the biplot approximation of $\mathbf{R}$ is given by

$$\mathbf{R}_{\alpha,k} = \mathbf{B}_{\alpha,k}\mathbf{B}_{\alpha,k}^\top = \mathbf{A}_k\mathbf{\Lambda}_k^\alpha\mathbf{\Lambda}_k^\alpha\mathbf{A}_k^\top , \tag{4.5}$$

where the choice $\alpha = \frac{1}{2}$ gives the best least-squares approximation to $\mathbf{R}$ of rank k. The standard biplots aim for low-dimensional data visualization, but the aim of the sBarse method is, primarily, a sparse and cheap loadings matrix. For this reason a wider interval for the power α is adopted. However, increasing the upper limit for α beyond 1 is not reasonable as this will result in one or very few PCs with poor approximation properties. As with standard biplots one can use a range of values α. Considering several $\alpha \in [0, 1]$ gives a list of biplots. Then, the most satisfying one can be chosen by the user or identified according to certain criterion.

It is natural to base this choice on the amount of the variance explained by the biplots and/or on their approximation power, measured by Gabriel (2002) as the goodness-of-fit of the biplot approximation $\mathbf{R}_{\alpha,k}$ to $\mathbf{R}$ using the RV-coefficient (Robert

and Escoufier, 1976):

$$\mathrm{RV}^2(\mathbf{R}, \mathbf{R}_{\alpha,k}) = \frac{\mathrm{trace}^2(\mathbf{R}\mathbf{R}_{\alpha,k})}{\mathrm{trace}(\mathbf{R}^2)\mathrm{trace}(\mathbf{R}_{\alpha,k}^2)} . \tag{4.6}$$

The RV values lie in the interval $[0, 1]$ and values close to 1 indicate better approximation.

4.2.2 Sparse biplots and sBarse components

The biplots considered in the previous section are standard, dense biplots. The sBarse method uses their sparse approximations, which are constructed following the approximation procedure outlined in Section 4.1.1. In fact, the construction described in Section 4.1.1 uses biplot $\mathbf{B}_\alpha$ with $\alpha = 0.5$, which is then sparsified into $\mathbf{V}$ to give the sBarse loadings matrix $\widetilde{\mathbf{V}}$. The sparse matrix $\widetilde{\mathbf{V}}$ is called a proper sBarse solution if it has first k ($\leq p$) non-zero columns and the last $p - k$ columns are identically zero. An improper solution is a solution which is not proper, i.e. a sparse matrix containing zero column(s) followed by non-zero columns. For example, the sBarse method applied to the Pitprop data with $\alpha = 0.5$ results in a proper sBarse solution $\widetilde{\mathbf{V}}_6$ with $k = 6$, given in Table 4.1.

The sBarse method, as introduced in Section 4.1.1, employs only one $\alpha = 0.5$. However, as with the biplots, one can use a range of values α. Thus, several $\mathbf{B}_\alpha, \alpha \in [0, 1]$ can be constructed to give a set of sparse matrices $\widetilde{\mathbf{V}}$. Once the set of sparse matrices $\widetilde{\mathbf{V}}$ is available the sBarse method excludes the improper solutions. Then, the most satisfying from the list of proper solutions is chosen by the user or identified according to a certain criterion.

In general, the small values of α lead to solutions with more sBarse components,

while the bigger α's correspond to fewer sBarse components. An interesting question is: how many α's to take to be sure that no valuable solution is missed? The answer is: not too many, because large intervals of α's correspond to a single set of sBarse components.

The essence of the sBarse algorithm is to produce a list of proper solutions $\tilde{\mathbf{V}}$ and rank them according to their explanatory power. The standard PCs are uncorrelated and their loadings matrix is orthogonal. However, the SCs from any sparse methods cannot satisfy these properties simultaneously. The loadings of the sBarse components are orthogonal to each other, but the components are correlated as $\mathbf{S}_k = \tilde{\mathbf{V}}_k^\top \mathbf{R} \tilde{\mathbf{V}}_k$ is not diagonal. As a consequence, the usual sum of variances trace($\mathbf{S}_k$) is usually too optimistic and not appropriate for SCs. Instead, Zou *et al.* (2006) introduced the adjusted variance for correlated SCs. Let $\mathbf{F}_k$ be the upper-triangular $k \times k$ factor of the Cholesky decomposition of $\mathbf{S}_k$, i.e. $\mathbf{S}_k = \mathbf{F}_k^\top \mathbf{F}_k$. Then, the squared elements on the main diagonal of $\mathbf{F}_k$ give the adjusted variances of the SCs.

The sBarse algorithm can be summarized as follows. For a set of values $\alpha \in [0, 1]$ the sBarse method finds biplot factors

$$\mathbf{B} := \mathbf{A}\mathbf{\Lambda}^\alpha \,, \tag{4.7}$$

where $\mathbf{A}$ is an orthogonal matrix and $\mathbf{\Lambda}^\alpha$ is diagonal. Then, the task is to find the set of proper sparse matrices $\mathbf{V}_k$ with elements from $\{-1, 0, 1\}$ which approximate $\mathbf{B}$. The number k of the sBarse components to retain may vary over the set of proper solutions. As the biplot factor $\mathbf{B}$ in (4.7) is a product, for interpretation purposes, the approximation $\mathbf{V}_k$ of $\mathbf{B}$ should also come as a product of an orthogonal matrix (of sparse loadings) multiplied by a diagonal matrix of "variances". That is why (in

Section 4.1), $\mathbf{V}_k$ is first normalized into $\tilde{\mathbf{V}}_k$ such that $\tilde{\mathbf{V}}_k^\top \tilde{\mathbf{V}}_k = \mathbf{I}_k$ and then is assigned to be the orthonormal term in the sparse biplot factor. The diagonal term in the sparse biplot factor could simply be formed by taking the variances of the new SCs (with sparse loadings $\tilde{\mathbf{V}}_k$), i.e. the main diagonal of $\mathbf{S}_k$. However, as the new SCs are correlated, it is reasonable to replace their variances by the corresponding adjusted ones. Then, the diagonal matrix containing the square root of the adjusted variances $\text{diag}(\mathbf{F}_k)$ is taken to be the second term in the sparse biplot factor. Thus, $\tilde{\mathbf{V}}_k\text{diag}(\mathbf{F}_k)$ is the sparse biplot factor and $\tilde{\mathbf{V}}_k$ is the sBarse loadings matrix. For a given value of α, the cumulative proportion of adjusted variances explained by the k sBarse components is given by

$$\text{adjvar}_\alpha = \frac{\text{trace}(\text{diag}^2(\mathbf{F}_k))}{p}. \tag{4.8}$$

It also seems reasonable to take into account the goodness-of-fit of the sparse biplots for each particular value of α. After substituting $\tilde{\mathbf{V}}_k\text{diag}(\mathbf{F}_k)$ into (4.6), the goodness-of-fit of the approximation of $\mathbf{R}$ is given by (Gabriel, 2002):

$$\begin{aligned}
\text{RV}_\alpha^2\left(\mathbf{R}, \tilde{\mathbf{V}}_k\text{diag}^2(\mathbf{F}_k)\tilde{\mathbf{V}}_k^\top\right) &= \frac{\text{trace}^2\left(\tilde{\mathbf{V}}_k^\top\mathbf{R}\tilde{\mathbf{V}}_k\text{diag}^2(\mathbf{F}_k)\right)}{\text{trace}(\mathbf{RR})\text{trace}\left(\tilde{\mathbf{V}}_k^\top\tilde{\mathbf{V}}_k\text{diag}^2(\mathbf{F}_k)\tilde{\mathbf{V}}_k^\top\tilde{\mathbf{V}}_k\text{diag}^2(\mathbf{F}_k)\right)} \\
&= \frac{\text{trace}^2\left(\text{diag}(\tilde{\mathbf{V}}_k^\top\mathbf{R}\tilde{\mathbf{V}}_k)\text{diag}^2(\mathbf{F}_k)\right)}{\text{trace}(\mathbf{\Lambda}^2)\text{trace}\left(\text{diag}^4(\mathbf{F}_k)\right)} \\
&= \frac{\text{trace}^2\left(\text{diag}(\mathbf{F}_k^\top\mathbf{F}_k)\text{diag}^2(\mathbf{F}_k)\right)}{\text{trace}(\mathbf{\Lambda}^2)\text{trace}\left(\text{diag}^4(\mathbf{F}_k)\right)}\ .
\end{aligned} \tag{4.9}$$

Thus, the proper solutions obtained from sBarse algorithm will be ranked according to the product of their cumulative adjusted variances (4.8) and their RV-coefficients (4.9). The best solution is the one with the maximum value of the product:

$$\text{adjvar}_\alpha \times \text{RV}_\alpha\ , \tag{4.10}$$

over different values of $\alpha \in [0, 1]$.

Let $\mathbf{A}$ and $\mathbf{\Lambda}$ be the matrices of eigenvectors and eigenvalues of $\mathbf{R}$. The sBarse procedure can be summarized in the following algorithm.

1) Set $\alpha = 0, \mathtt{max}_\alpha = 0$ and a discretization step Δ, say $\Delta = .02$.

2) Compute $\mathbf{B} = \mathbf{A}\mathbf{\Lambda}^\alpha$.

3) Obtain the sparse matrix $\mathbf{V}$ from $\mathbf{B}$ (as in Section 4.1), i.e.:

$$v_{ij} = \begin{cases} \frac{b_{ij}}{|b_{ij}|}\ , & \text{if } |b_{ij}| = \max(|b_{i1}|, |b_{i2}|, ..., |b_{ip}|) \\ 0\ , & \text{otherwise} \end{cases} .$$

4) Check that $\mathbf{V}$ is a proper solution, i.e. all first k ($\leq p$) columns of $\mathbf{V}$ are non-zero. If not, go to 13).

5) If yes, cut off the last $p - k$ columns of $\mathbf{V}$ to form $\mathbf{V}_k$.

6) Check that this $\mathbf{V}_k$ has not been found before yet.

7) If it has, go to 13).

8) If $\mathbf{V}_k$ is new, normalize $\mathbf{V}_k$ in $\widetilde{\mathbf{V}}_k$ such that $\widetilde{\mathbf{V}}_k^\top \widetilde{\mathbf{V}}_k = \mathbf{I}_k$.

9) Compute the Cholesky decomposition: $\widetilde{\mathbf{V}}_k^\top \mathbf{R} \widetilde{\mathbf{V}}_k = \mathbf{F}_k^\top \mathbf{F}_k$.

10) Compute the cumulative proportion of adjusted variances: $\mathtt{adjvar} = \text{trace}(\text{diag}^2(\mathbf{F}_k))/p$.

11) Compute the RV-coefficient (RV) using (4.9).

12) Compute $\mathtt{max}_\alpha = \mathtt{adjvar} \times \mathtt{RV}$. Compare the new $\mathtt{max}_\alpha$ with the old one, and keep the value of α, say $\alpha_{\max}$, for which $\mathtt{max}_\alpha$ is the largest.

13) Increment α by Δ, i.e., $\alpha \leftarrow \alpha + \Delta$. If $\alpha \leq 1$, go to step (2); otherwise, stop the algorithm. $\widetilde{\mathbf{V}}_k$ corresponding to $\alpha_{\max}$ is the matrix of sBarse loadings.

It is possible that the best value of α_{max} can be missed if the value of Δ is not small enough. However, the smaller the value of Δ, the slower the algorithm, especially if p is large (such as in the gene expression data). As an alternative, an improved value of α_{max} can be searched in a neighbourhood of the current value by continuously narrowing the search interval and repeating the above algorithm. Suppose that the algorithm is applied first on the interval [0,1] and the best solution is obtained for $\alpha^{(1)}_{\mathbf{max}}$. Let $LL^{(1)}$ and $UL^{(1)}$ denote the lower and the upper limits of the range of α, so for the first stage $LL^{(1)} = 0$ and $UL^{(1)} = 1$. For the second stage, the limits are updated as

$$LL^{(2)} \leftarrow .5(\alpha^{(1)}_{\mathbf{max}} + LL^{(1)}) \qquad \text{and} \qquad UL^{(2)} \leftarrow .5(\alpha^{(1)}_{\mathbf{max}} + UL^{(1)})$$

and repeat the sBarse algorithm on the updated interval $[LL^{(2)}, UL^{(2)}]$. Then check if the value of the resulting $\mathbf{max}^{(2)}_{\alpha}$ changes. The value of $\Delta^{(i)}$ at the ith repetition (stage) of the algorithm can be set to some function of the difference between $LL^{(i)}$ and $UL^{(i)}$, say,

$$\Delta^{(i)} = .1(UL^{(i)} - LL^{(i)}) \, .$$

At the $(i+1)$th stage $(i = 1, 2, \cdots)$, update the interval as

$$LL^{(i+1)} \leftarrow .5(\alpha^{(i)}_{\mathbf{max}} + LL^{(i)}) \qquad \text{and} \qquad UL^{(i+1)} \leftarrow .5(\alpha^{(i)}_{\mathbf{max}} + UL^{(i)})$$

and repeat the sBarse algorithm. If $\mathbf{max}^{(i)}_{\alpha} = \mathbf{max}^{(i+1)}_{\alpha}$, then stop the algorithm and use the matrix of sBarse components corresponding to $\alpha_{\mathbf{max}} = \alpha^{(i)}_{\mathbf{max}}$.

If the raw data matrix $\mathbf{X}$ is available, there is no need to form the sample correlation matrix $\mathbf{R}$ in the sBarse algorithm. Indeed, consider the QR decomposition $\mathbf{X}\tilde{\mathbf{V}}_k = \mathbf{QT}$, where $\mathbf{Q}$ is an $n \times k$ orthonormal matrix, and $\mathbf{T}$ is an $k \times k$ upper-triangular. [Note that $\tilde{\mathbf{V}}_k$ can be obtained from the SVD of $\mathbf{X}$.] As shown by Zou *et al.* (2006),

the adjusted variances of the new SCs are given by the squared elements on the main diagonal of $\mathbf{T}$. If the diagonal matrix containing the adjusted variances is denoted by $\text{diag}^2(\mathbf{T})$, then the required sparse biplot is $\tilde{\mathbf{V}}_k\text{diag}(\mathbf{T})$, and the RV-coefficient (4.9) is found by simply making $\mathbf{F}_k \equiv \mathbf{T}_k$.

4.3 Further application

In this section, some more details are given for the sBarse solution of the Pitprop data (Jeffers, 1967) considered above. The results are compared with other existing sparse solutions. Next, simulated data are used to show the performance of the proposed method. Finally, a real gene expression data set (Chin *et al.*, 2006) is considered, where the sBarse solutions are compared with sparse solutions obtained by other methods.

The Pitprop data

For the Pitprop data (continued from Section 4.1.2), there are 29 proper solutions (out of 51) obtained by the sBarse algorithm with $\alpha \in [0, 1]$ and a step size of .02. For many values of α, identical sBarse components are found. The sBarse algorithm checks and omits them, i.e. the recalculation of their variances, adjusted variances and RV-coefficients is not needed. It is found that for $\alpha = 0.36$ the algorithm produces the best proper solution with six sBarse components (the last six columns of Table 4.1), accounting for 76.8% of the total unadjusted variance and 73.3% of the total adjusted variance. This value of α is not uniquely defined; other values of α, say $\alpha = 0.4$ or $\alpha = 0.5$, also result in the same solution. Hence, the best solution corresponds to an interval of α values. According to the value of the product of total adjusted

variance explained and the RV coefficient, the best solution has 6 sBarse components, whose characteristics are reported in Table 4.2. The same best solution is obtained at $\alpha = 0.35$ using the interval-narrowing approach with less computation time.

Note that all existing methods for sparse PCA applied to the Pitprop data (d'Aspremont *et al.*, 2007; Farcomeni, 2009; Jolliffe *et al.*, 2003; Moghaddam *et al.*, 2006; Zou *et al.*, 2006) a priori employ the first six SCs explaining a reasonable portion of the original variance. In contrast, the sBarse method *finds* the appropriate number of SCs, which happens to be 6. The choice of 4 sBarse components would correspond to Kaiser's criterion to retain the first four PCs with variances greater than 1 (explaining 73.97% of the total variation).

Table 4.2: *Proper sBarse components for the Pitprop data for* $\alpha \in [0, 1]$ and step .02.

Sol	α	RV	Var	Adj	RV×Adj	# sBarse comp.
1	.36	.8580	.7684	.7325	.6285	6
2	.68	.8233	.5938	.5910	.4866	4
3	.92	.7424	.5590	.5497	.4081	4
4	.94	.5829	.4801	.4426	.2580	4
5	.96	.5339	.4428	.4294	.2293	4
6	1.00	.6109	.5016	.4857	.2967	4

This example also shows that the sBarse method does not produce sparse loadings for any $k = 1, 2, ..., p$. This is a disadvantage of the method as it might be necessary to have a sparse solution with particular number of components, which the method cannot produce. In the same time, this can be viewed as an advantage of the method as it reduces the freedom of the choice of the proper number of components to retain.

Table 4.3 gives the loadings of the first three SCs and the corresponding cumulative variances (CV), adjusted variances (CAV) and number of zero-loadings (0*s*) found by

the several methods. Most of them produce 4th, 5th and 6th SCs with a single non-zero (unit) loading. The values in the table are collected from the original papers, where available, otherwise they have been computed by the author. The abbreviations are: SPC – simple principal components (Vines, 2000), SPCA – sparse principal component analysis (Zou *et al.*, 2006), SCoTLASS – Simplified Component Technique-LASSO (Jolliffe *et al.*, 2003) with $\tau = (2.5, 1.5, 1.5, 1.01, 1.01, 1.01)$, DSPCA – direct sparse PCA (d'Aspremont *et al.*, 2007), ESPCA – exact sparse PCA (Moghaddam *et al.*, 2006), SCA – simple component analysis (Rousson and Gasser, 2004) and IDR – interpretable dimensionality reduction (Chipman and Gu, 2005) with H, C, and S for homogeneity, contrasts and sparsity constraints respectively.

The solutions produced by SPC (Vines, 2000), IDR H and C (Chipman and Gu, 2005) and SCoTLASS (Jolliffe *et al.*, 2003) are not sparse. The worst solution seems to be the SCA one (Rousson and Gasser, 2004) which explains only 47% of the adjusted variance (and 66% for all six sparse components, also not much). The ESPCA solution (Moghaddam *et al.*, 2006) is the sparsest one, but explains only 49% of the adjusted variance. DSPCA (d'Aspremont *et al.*, 2007) is a bit less sparse, but also not quite satisfying with 50% adjusted variance. The sBarse solution is the sparsest one of the three remaining with 53% explained adjusted variance. The SPCA (Zou *et al.*, 2006) explains 55% adjusted variance at the price of 5 more non-zeros compared to the sBarse solutions. The IDR solution (Chipman and Gu, 2005) with sparsity constraint (with $\eta = .9$) explains 56% adjusted variance, being less sparse than the sBarse solution. However, the IDR solution lacks orthogonality, which devalues its quality as the sBarse and SPCA loadings are exactly orthonormal. An additional weakness of the IDR and SPCA solutions is that there are variables contributing to more than one SC. In fact,

Table 4.3: *SC loadings and variances explained by different methods, Pitprop data.*

Empty cells have zero values.

Method	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	CV	CAV	0s
sBarse 1	-.41	-.41					-.41	-.41	-.41	-.41				29	29	7
sBarse 2			.71	.71										43	43	11
sBarse 3					.71	.71								54	53	11
SPC 1	.32	.32			.32	.32	.32	.32	.32	.32		-.32	-.32	28	28	3
SPC 2	.44	.44	.22	.22	-.44				.22		.22	.22	.44	47	46	4
SPC 3	.08	.08	-.37	-.37	-.21	-.46	-.37	.33	.04	.33		-.29	.12	61	59	1
SPCA 1	-.48	-.48			.18		-.25	-.34	-.42	-.40				28	28	6
SPCA 2			.79	.62				-.02				.01		42	42	9
SPCA 3					.64	.59	.49						-.02	57	55	9
SCoTLASS 1	-.48	-.49				-.11	-.38	-.25	-.38	-.41				30	30	6
SCoTLASS 2			.70	.71		.06				-.02		.01		45	44	8
SCoTLASS 3	-.06	-.09	-.02		.02	.22	.13						-.96	55	54	6
DSPCA 1	-.56	-.58					-.26	-.10	-.37	-.36				27	27	7
DSPCA 2			.71	.71										42	40	11
DSPCA 3						.79	.61						-.01	56	50	10
ESPCA 1	-.48	-.49					-.41		-.42	-.43				26	26	8
ESPCA 2			.71	.71										41	40	11
ESPCA 3						.81	.58							55	49	11
SCA 1	.45	.45						.45	.45	.45				25	25	8
SCA 2					.50	.50	.50						.50	35	34	9
SCA 3			.71	.71										49	47	11
IDR H1	-.38	-.38				-.38	-.38	-.38	-.38	-.38				30	30	6
IDR H2	-.30	-.30	-.30	-.30	.30		.30	.30		.30	-.30	-.30	-.30	47	46	2
IDR H3	-.33	-.33		.33	.33	.33	.33	-.33	-.33				-.33	61	57	4
IDR C1	-.15	-.15	-.15	-.15	-.15	-.15	-.15	-.15	-.15	-.15	.51	.51	.51	16	16	0
IDR C2	-.23	-.23	-.23	-.23	.40		.40	.40		.40	-.23	-.23	-.23	32	24	2
IDR C3	-.30	-.30		.37	.37	.37	.37	-.30	-.30				-.30	45	36	4
IDR S1	-.42	-.42				-.30	-.42	-.31	-.37	-.39				31	31	6
IDR S2			-.69	-.58								-.44		45	45	10
IDR S3				.43	.58	.57							-.39	59	56	9

such overlapping effect is present in all solutions except the sBarse one. The sBarse solution of the Pitprop data seems to be the best one with respect to overall sparseness, ease of interpretation and goodness-of-fit.

The classical PCs are both orthogonal and uncorrelated. The SCs cannot preserve these two features simultaneously. The orthogonality of the SCs is maintained exactly only by the sBarse method, SCoTLASS and SPCA. The rest of the methods maintain the solutions' orthogonality only approximately, with the IDR (Chipman and Gu, 2005) deviating most. The correlations among the SCs obtained by the three best sparse solutions of the Pitprop are given in Table 4.4. The correlation structures of the sBarse and SPCA solutions are quite similar.

Table 4.4: *Correlations among six SCs from three methods for the Pitprop data*

	sBarse $\alpha = .4$					SPCA					IDR Sparse ($\eta = .9$)				
Var	x_1	x_2	x_3	x_4	x_5	x_1	x_2	x_3	x_4	x_5	x_1	x_2	x_3	x_4	x_5
x_2	.16					-.17					.11				
x_3	-.26	-.19				-.33	.13				-.39	-.35			
x_4	.03	-.13	-.08			-.00	-.14	.10			-.26	.13	.17		
x_5	-.24	.20	.07	-.03		-.20	-.22	.14	.03		-.12	-.27	.09	-.05	
x_6	.15	-.07	-.33	.01	-.18	.08	.08	-.39	-.01	-.18	.09	-.08	.16	-.29	.07

Simulated data

Here, we test the performance of the sBarse method using artificial data, generated by one of the models originally considered by Jolliffe (1972). The model is constructed in such a way that 10 variables x_l are linear combinations of 10 independent standardized normal variables, z_i, as in Table 4.5.

The model is constructed in such a way that the variables, x_l, fall into groups. The

Table 4.5: *Formulae for generating artificial data (Jolliffe, 1972)*

Variable	Variate comb.	Variable	Variate comb.
x_1	z_1	x_6	$2z_4 + 0.75z_5 + 1.5z_6$
x_2	z_2	x_7	z_7
x_3	$z_2 + z_3$	x_8	$z_7 + 0.5z_8$
x_4	z_4	x_9	$2z_7 + 0.5z_8 + z_9$
x_5	$z_4 + 0.75z_5$	x_{10}	$3z_7 + z_8 + z_9 + z_{10}$

variables in each group are linear combinations of the same underlying z_i, whereas the variables from different groups are independent. The 10 variables fall into 4 groups: $\{x_1\}$, $\{x_2, x_3\}$, $\{x_4, x_5, x_6\}$, and $\{x_7, x_8, x_9, x_{10}\}$.

The correlation matrix of the variables $x_1, ..., x_{10}$ is calculated for 100 generated observations. The correlation between variables from different groups is very small (and is assumed to be zero), while the correlation between variables from the same group is large.

The first four PCs of the correlation matrix of the simulated data account for 92.5% of the total variation. This suggests that it suffices to find the first four SCs. Application of the sBarse algorithm to the correlation matrix, with narrowing search-interval, results in two proper solutions. The best solution has four sBarse components at $\alpha = 0.15$ and $\mathbf{max}_\alpha = 0.91$. Table 4.6 gives the loadings and cumulative variances of the four PCs and the best sBarse components. Each of the four sBarse components reconstructs correctly the corresponding original PCs. The four sBarse components account for 91.9% of the total adjusted variance in the original data. This percentage

is almost the same as that of the ordinary PCs, but the sBarse components are much easier to interpret than the PCs.

Table 4.6: *Loadings and cumulative adjusted variances* ($\%Cvar_{adj}$) *of the first four PCs and the corresponding sBarse components, simulated data. Empty cells have zero values.*

	Principal Components				sBarse Components			
Variable	1	2	3	4	1	2	3	4
1	-.004	-.047	-.119	.985				1
2	-.116	.129	-.672	-.155			-.707	
3	-.093	.102	-.696	.001			-.707	
4	.040	.572	.084	.032		.577		
5	.039	.570	.125	.049		.577		
6	.003	.563	.053	.035		.577		
7	-.491	.010	.105	.021	-.500			
8	-.492	-.001	.064	-.024	-.500			
9	-.494	.034	.069	.010	-.500			
10	-.498	.001	.069	.028	-.500			
$\%Cvar_{adj}$	38.4	65.1	82.6	92.5	37.9	64.3	82.0	91.9

Gene expression data, $p \gg n$

The sBarse method can be applied to a large data set (such as the microarray gene expression) where the number of variables (p) is much larger than the number of samples (n). Here, we use a real breast cancer gene expression data set first considered by Chin *et al.* (2006) and publicly available from `http://icbp.lbl.gov/breastcancer/`.

Witten *et al.* (2009) used this data set to illustrate the penalized matrix decomposition method for obtaining sparse PCs. They analyze 19,672 gene expression measurements on 89 samples. For computational reasons, Witten *et al.* (2009) used a subset

of the data consisting of the 5% of genes with highest variance. For comparison, we also use the same subset of data, say $\mathbf{X}$, which consists of $p = 984$ genes (variables) and $n = 89$ samples.

The data are first standardized and the SVD used to obtain the matrix of principal component loadings, $\mathbf{A}$, and the corresponding matrix of singular values, $\mathbf{L}$. Since $p \gg n$ and the data are mean centered, the SVD results in $(n-1)$ nonzero singular values. As a result, the sBarse method is based on the $p \times (n-1)$ matrix of loadings $\mathbf{A}_{n-1}$ and the $(n-1) \times (n-1)$ diagonal matrix of eigenvalues $\mathbf{\Lambda}_{n-1} = \mathbf{L}^2_{n-1}$. As the method depends on the nonzero singular values, the maximum number of sBarse components k that one can obtain is $n-1$. (In general, $k \leq \min\{n-1, p\}$, whether $n > p$ or $n \ll p$.)

The sBarse method applied to the breast cancer gene expression data resulted in 88 sBarse components. This is the maximum number of sBarse components for this data set, as $n = 89$. The best sBarse solution is obtained when $\alpha = 0.125$, at which the RV-coefficient is .3337, and the cumulative unadjusted and adjusted variances explained by the 88 components are 27.4% and 19.4%, respectively. The number of nonzero-loading genes in a sBarse component ranges from 1 to 92, and each of the 984 genes gets a nonzero loading in only one component. The number of nonzero-loading genes in the sBarse components generally decreases with decreasing percentage of variances explained by the components. As the bar chart in Figure 4.1 shows, the majority of the genes are included in the first few sBarse components. The user can choose the required number of components depending on the cumulative percentage of variances and the level of sparsity.

We compare the sBarse components with the SCs obtained by Witten *et al.* (2009),

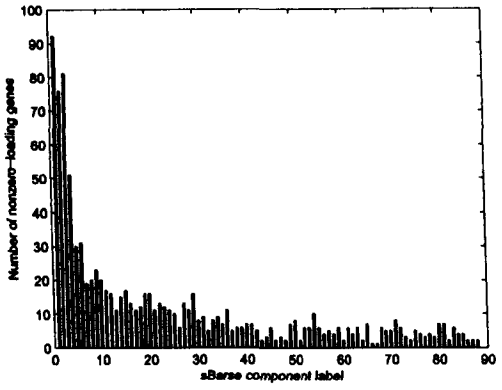

Figure 4.1: *Number of nonzero-loading genes in each of the 88 sBarse components, displayed in decreasing order of the percentage of variance explained by the components*

abbreviated hereafter as the SPC method. The two methods are compared with respect to the level of sparsity (number of nonzero-loading genes) of the components and the cumulative percentage of adjusted variances explained by the sparse components.

For a fair comparison between SPC and sBarse, some adjustments are employed. The sBarse method cannot control the level of sparsity in a SC, while the SPC method can do this explicitly by requiring a particular sum of absolute values of loadings in a SC (via the input argument **sumabsv** in the R function SPC). Also, by construction, the sBarse method results in components involving non-overlapping genes. The SPC method lacks this feature. However, **sumabsv** can be obtained from the sBarse components and the SPC components can be made as non-overlapping as possible, so that both the SPC and sBarse methods are put on a similar footing for a fair comparison. This can be accomplished using the following procedures:

a. Run the sBarse algorithm and compute $c_1, \ldots, c_m$ where c_i is the sum of absolute values of the p elements in the ith sBarse component with the ith largest variance.

b. Run the SPC algorithm with **sumabsv** $= c_1$ in order to get component 1. If $\mathbf{v}_1$ denotes the first SPC, computed based on the data matrix $\mathbf{X}_1 = \mathbf{X}$, then the second sparse component $\mathbf{v}_2$ is computed on the residual data matrix $\mathbf{X}_2 \equiv \mathbf{X}_1 - \mathbf{X}_1\mathbf{v}_1\mathbf{v}_1^\top$. Then, perform SPC on $\mathbf{X}_2$ with **sumabsv** $= c_2$ to get component 2.

c. Repeat this procedure until a required number k' $(\leq k)$ of SPCs has been obtained. In general, the ith SPC $\mathbf{v}_i$ is computed based on the residual data matrix $\mathbf{X}_i \equiv \mathbf{X}_{i-1} - \mathbf{X}_{i-1}\mathbf{v}_{i-1}\mathbf{v}_{i-1}^\top$ for $i = 2, \ldots, k'$.

For the breast cancer gene expression data, the c_i's computed from the first 25 sBarse components are used to get 25 SPCs. The values of **sumabsv** are generally decreasing from 9.5917 (for the first sBarse component) to 3.1625 (for the 25th sBarse component) (see Figure 4.1).

The plot on the left hand side of Figure 4.2 gives the number of nonzero-loading genes in each of the first 25 SCs for both the sBarse and SPC methods. For both methods, the number of nonzero-loading genes generally decreases (and, hence, the level of sparsity increases) with decreasing variance explained. However, the SCs from the sBarse method are sparser than the corresponding SCs from the SPC method. In other words, at a given value of **sumabsv**, a sBarse component tends to have a smaller number of nonzero-loadings, possibly each with larger absolute values, while an SPC component tends to have larger number of nonzero-loading genes, each with possibly smaller absolute values. Thus, the sBarse method seems superior to the SPC method in simplifying interpretation of the components.

The plot on the right hand side of Figure 4.2 shows the cumulative proportion of adjusted variances explained by the first 25 SCs for both methods. The sBarse

components explain 11% less variance than the corresponding SPC ones. Note, that the first three sBarse and SPC components are equally informative.

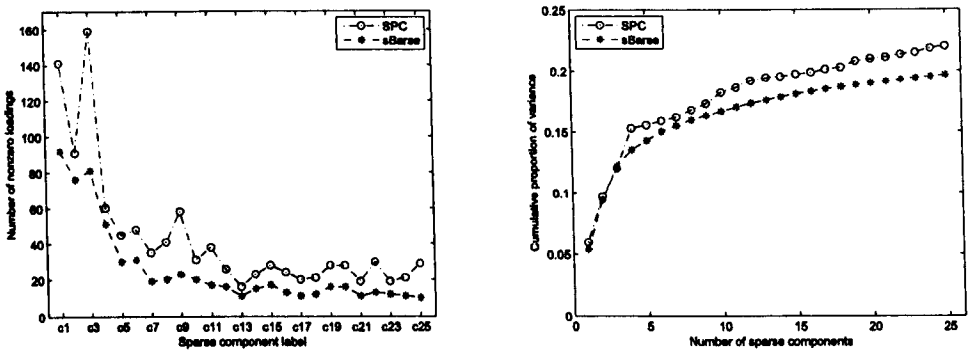

Figure 4.2: *The first 25 sparse components from sBarse and SPC methods for breast cancer data - (left) number of nonzero-loadings, and (right) cumulative adjusted variances explained*

Components with non-overlapping genes may simplify interpretation of the components. The sBarse components possess this property, while it is not guaranteed by the SPC method.

For large data (such as gene expression), the size of the step length in searching for the best $\alpha \in [0,1]$ is crucial. In general, a shorter step length can result in a better solution, but requires more computational time. Hence, it is recommended to consider a compromise between the computation time, the required level of sparsity, and the total variance explained when choosing the step length. It turns out, the computation time can be reduced considerably by starting with a larger step length and then reducing it, repeating the sBarse algorithm only in a neighborhood of the current best value of α (see Section 4.2.2). For the breast cancer gene expression data, it takes only 4.45 minutes to give the sBarse solution, on an Intel(R) Pentium 4

desktop computer with 3.2GHz CPU and .99 GB of Ram.

4.4 Summary

In this chapter, a simple and fast approach to interpretable PCs is proposed. Simplicity in the interpretation of a component is related to its level of sparsity, which is inferred from the number of zero-loading variables. The objective is to make a component as sparse as possible so that it is easily interpretable without losing much information contained in the original variables. A nice feature of the method is that it is clearly aimed at PCA and not factor analysis, because it keeps enough components to 'explain' all the variables.

The technique involves a biplots approach to matrix approximation, and hence referred to as sparse biplots (sBarse) component analysis. Like the ordinary PCA, it requires us compute the eigenvalues and eigenvectors of a data or correlation matrix. An additional requirement is the estimation of α. But, as $\alpha \in [0, 1]$ and, intervals of α values correspond to a single solution, the choice of the value of α may not be considered as a serous problem.

The 'best' sBarse solution is chosen based on a criterion involving the product of the cumulative proportion of adjusted variances explained by the sBarse components and the goodness-of-fit of the biplot approximation to the correlation matrix, given by the RV-coefficient. The larger the value of the criterion is the better the solution.

Results of different examples show that the sBarse method produces k ($\leq p$) sparse components, each of which are sparser than and/or explain at least as high proportion of adjusted variance as those obtained by other similar approaches proposed in

literature.

Chapter 5

Clustering approach to interpretable principal components

5.1 Introduction

In Chapter 3, we outlined a variety of approaches proposed for simplifying the interpretation of PCs . Rotation to simple structure is the oldest approach, initially designed in factor analysis and later adapted to PCA. It aims to make the rotated components as interpretable as possible. However, the belief that a rotated component has its absolute loadings near 1 or 0, while avoiding intermediate values, is not usually true and makes interpretation ambiguous. On the other hand, most of the modern simplifying approaches are designed to set or drive some of the component loadings to exact zeros in order to make the components interpretable. This trend was initiated by Hausman (1982), who constrained the PC loadings to the set of three values, $\{-1, 0, 1\}$. The SCoTLASS problem (Jolliffe *et al.*, 2003) requires maximization of the standard PCA objective function subject to an additional LASSO constraint. It triggered a series

of papers where alternative methods were proposed. The sparse principal component analysis (Zou *et al.*, 2006) uses a constrained technique (thresholding) to drive some of the component loadings to exact zeros. Similarly, d'Aspremont *et al.* (2007) propose a cardinality-constrained objective function for the same purpose. Chipman and Gu (2005) introduce three types of "interpretable" components, each corresponding to homogeneity, contrast and sparsity constraints.

Interpretation of a PC can be associated with the level of sparsity of the component, measured by the number of zero (or non-zero) loadings. The larger the number of zero loadings, the sparser the component and the easier the interpretation. Unfortunately, components resulting from some of the above approaches are not sparse enough, and some of the sparse components might still not be easily interpretable.

On the other hand, simple component analysis (SCA) as proposed by Rousson and Gasser (2004) involves clustering of variables. They approximate the first k ($\leq p$) PCs by a mixture of b 'block' and $(k-b)$ 'difference' components, in which the block components are computed from the correlation matrices of each cluster. The argument behind SCA is that the block components are easier to interpret than the difference components, and hence aims to increase the number of block components. Vichi and Saporta (2009) propose a constrained PCA approach which aims to simultaneous clustering of observations and partitioning of variables. The set of variables in each partition helps to make a 'disjoint' PC with maximum variance. Jolliffe (2002) discusses the possibility of deducing approximate PCs from the patterns of correlation matrix, which requires the detection of well-defined groups (clusters) of variables. However, such patterns may not be easily visible in many real correlation matrices.

There is a genuine connection between PCA and cluster analysis. Suppose that

the first k PCs of a matrix account for the majority of the variation in the original data. Then, one possible measure of dissimilarity between pairs of observations is the Euclidean distance in the k-dimensional subspace defined by these PCs. However, it has been pointed out (Jolliffe, 2002, p.211) that there is no real advantage in using this measure instead of the Euclidean distance in the original p-dimensional space. In addition, Yeung and Ruzzo (2001) used PCA for clustering observations and argue that clustering with the PCs instead of the original observations does not necessarily improve, and often degrades, cluster quality. Another connection is that PCA can help to identify the presence of clusters of variables and hence can be considered as a competitor to cluster analysis. When variables fall into well-defined clusters, then there will be one PC with high variance and one or more PCs with low variance associated with each cluster, except in the case where a cluster has only one variable.

In this chapter, we propose a cluster-based approach for constructing interpretable principal components (IPCs). The p variables are first grouped into k 'best' clusters, each with q_j variables ($j = 1, \ldots, k$), based on a given criterion, and then the jth IPC is constructed from the correlation matrix of the jth cluster. Thus, the jth IPC contains q_j nonzero loadings corresponding to the variables in the cluster and $(p - q_j)$ exact-zero loadings corresponding to the variables outside the cluster, with $\sum_j^k q_j = p$. The resulting k IPCs are assumed to approximate the first k PCs with respect to the cumulative percentage of adjusted variance (Zou *et al.*, 2006) and the structure of the component loadings. For this purpose, a new weighted-variance clustering method is proposed. In general, the IPC algorithm involves two stages – grouping the p variables into k non-overlapping clusters, and constructing the IPCs from the correlation matrix of each cluster. Due to the design of the clustering algorithm, which requires p weights

(Section 5.3), we explicitly assume in this chapter that $n > p$. A cluster-based method for the case $p \gg n$ will be considered in Chapter 6.

Vigneau and Qannari (2003) developed similar procedure to IPC, but they use a different criterion and, unlike our method, fix the number of clusters a priori. The general idea of the IPC method also has some similarity with the computation of the 'block' components by Rousson and Gasser (2004), and the 'disjoint' PCs by Vichi and Saporta (2009), but the methods are quite different with respect to the simplicity of the algorithm involved and the interpretability of the resulting components.

The chapter is organized as follows. Section 5.2 gives the motivation, including two simple motivating examples. In Section 5.3, we propose the new clustering method. Section 5.4 is devoted to the construction of the IPCs. Applications of the method to simulated and to real data sets are given in Section 5.5. The chapter is briefly summarized in Section 5.6.

5.2 Motivation

A clustering approach to IPCs is motivated by the specific form of the eigenvalue decomposition (EVD) of a block-diagonal correlation matrix. Let $\mathbf{R}$ be the following $p \times p$ block-diagonal correlation matrix:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{q_1} & \mathbf{0}_{q_1 \times q_2} & \cdots & \mathbf{0}_{q_1 \times q_k} \\ \mathbf{0}_{q_2 \times q_1} & \mathbf{R}_{q_2} & \cdots & \mathbf{0}_{q_2 \times q_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{q_k \times q_1} & \mathbf{0}_{q_k \times q_2} & \cdots & \mathbf{R}_{q_k} \end{bmatrix}, \tag{5.1}$$

where each block $\mathbf{R}_{q_i}$ is a $q_i \times q_i$ correlation matrix and $\sum_{i=1}^{k} q_i = p$. Then, the eigenvalues of $\mathbf{R}$ are solutions of the following equation:

$$f(\lambda) = \det(\mathbf{R} - \lambda \mathbf{I}_p) = \prod_{i=1}^{k} \det(\mathbf{R}_{q_i} - \lambda_i \mathbf{I}_{q_i}) = 0 \ ,$$

i.e. the eigenvalues of $\mathbf{R}$ can be found by solving k smaller eigenvalue problems for $\mathbf{R}_{q_1}, ..., \mathbf{R}_{q_k}$ (Horn and Johnson, 1985, p.24). Let $\mathbf{R}_{q_i} = \mathbf{A}_{q_i} \mathbf{L}_{q_i}^2 \mathbf{A}_{q_i}^\top$ denote the EVD of $\mathbf{R}_{q_i}$. Then, after substitution in (5.1) one finds that

$$\mathbf{R} = \begin{bmatrix} \mathbf{A}_{q_1} \mathbf{L}_{q_1}^2 \mathbf{A}_{q_1}^\top & \mathbf{0}_{q_1 \times q_2} & \cdots & \mathbf{0}_{q_1 \times q_k} \\ \mathbf{0}_{q_2 \times q_1} & \mathbf{A}_{q_2} \mathbf{L}_{q_2}^2 \mathbf{A}_{q_2}^\top & \cdots & \mathbf{0}_{q_2 \times q_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{q_k \times q_1} & \mathbf{0}_{q_k \times q_2} & \cdots & \mathbf{A}_{q_k} \mathbf{L}_{q_k}^2 \mathbf{L}_{q_k}^\top \end{bmatrix} = \mathbf{A} \mathbf{L}^2 \mathbf{A}^\top \ , \tag{5.2}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{q_1} & \mathbf{0}_{q_1 \times 1} & \cdots & \mathbf{0}_{q_1 \times 1} \\ \mathbf{0}_{q_2 \times 1} & \mathbf{A}_{q_2} & \cdots & \mathbf{0}_{q_2 \times 1} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{q_k \times 1} & \mathbf{0}_{q_k \times 1} & \cdots & \mathbf{A}_{q_k} \end{bmatrix} \ , \tag{5.3}$$

with $\mathbf{A}_{q_i}^\top \mathbf{A}_{q_i} = \mathbf{A}_{q_i} \mathbf{A}_{q_i}^\top = \mathbf{I}_{q_i}$ for each i, which implies $\mathbf{A}^\top \mathbf{A} = \mathbf{A} \mathbf{A}^\top = \mathbf{I}_p$, and

$$\mathbf{L}^2 = \begin{bmatrix} \mathbf{L}_{q_1}^2 & \mathbf{0}_{q_1 \times q_2} & \cdots & \mathbf{0}_{q_1 \times q_k} \\ \mathbf{0}_{q_2 \times q_1} & \mathbf{L}_{q_2}^2 & \cdots & \mathbf{0}_{q_2 \times q_k} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{q_k \times q_1} & \mathbf{0}_{q_k \times q_2} & \cdots & \mathbf{L}_{q_k}^2 \end{bmatrix} \ . \tag{5.4}$$

Thus, PCA of a block-diagonal correlation matrix results in a sparse loadings matrix (5.3). This feature was partially exploited by Rousson and Gasser (2004) for small p. In this chapter, this feature is used to construct *orthogonal* sparse components.

In the remaining part of this section, two data sets, one hypothetical and the other real, are considered for motivating the IPCs method. The correlation matrix of the hypothetical data set is constructed in such a way that the corresponding PCs are sparse. The example helps to grasp intuitively the idea of the IPC method. The second (real) data set is taken from McCabe (1984) and will be used throughout the rest of the chapter for demonstration.

Motivating example 1

Consider a hypothetical correlation matrix $\mathbf{R}$ of five variables, x_1, x_2, x_3, x_4 and x_5, with two well-defined groups, $\{x_1, x_2\}$ and $\{x_3, x_4, x_5\}$, as shown in Table 5.1. The correlation coefficient between a variable from one group and a variable from another group is zero.

Table 5.1: *Hypothetical correlation matrix* $\boldsymbol{R}$ *and its PCs*

	R					PC loadings				
Variable	x_1	x_2	x_3	x_4	x_5	PC1	PC2	PC3	PC4	PC5
x_1	1	.75	0	0	0	-.7071	0	0	0	-.7071
x_2		1	0	0	0	-.7071	0	0	0	.7071
x_3			1	.43	.17	0	-.6022	.4798	.6380	0
x_4				1	.27	0	-.6478	.1734	-.7418	0
x_5					1	0	-.4666	-.8601	.2064	0
Variance						1.75	1.5942	.8507	.555	.25

The last five columns of Table 5.1 gives the PCs of $\mathbf{R}$. The effect on the PC loadings of the zero-valued correlations is clearly noticeable from the exact-zero loadings. Each of the five PCs are sparse, in that each PC gets nonzero-loadings only for the variables in one group. The two nonzero-loading variables for PC1 and PC5 correspond to

$\{x_1, x_2\}$, while the three nonzero-loading variables for PC2, PC3, and PC4 correspond to $\{x_3, x_4, x_5\}$. Thus, the interpretation of each PC involves only the nonzero-loading variables in the corresponding component.

On the other hand, the nonzero-loadings of the sparse principal components in Table 5.1 can be obtained directly from the correlation matrices of each cluster of variables. To see this, let $\mathbf{R}_1$ and $\mathbf{R}_2$ denote the correlation matrices of the variables in $\{x_1, x_2\}$ and $\{x_3, x_4, x_5\}$, respectively. These matrices, together with their corresponding PCs, are shown in Table 5.2. Note from the table that the loadings and variances (eigenvalues) of the two PCs of $\mathbf{R}_1$ are exactly the same as the nonzero-loadings and variances of PC1 and PC5 of $\mathbf{R}$. Similarly, the loadings and variances of the three PCs of $\mathbf{R}_2$ are the same as the nonzero-loadings and variances of PC2, PC3, and PC4 of $\mathbf{R}$.

Table 5.2: *Hypothetical correlation submatrices $\boldsymbol{R}_1$ and $\boldsymbol{R}_2$ and their PCs*

	$\mathbf{R}_1$		PC loadings			$\mathbf{R}_2$			PC loadings		
Variable	x_1	x_2	$PC1_1$	$PC2_1$	Variable	x_3	x_4	x_5	$PC1_2$	$PC2_2$	$PC3_2$
x_1	1	.75	-.7071	-.7071	x_3	1	.43	.17	-.6022	.4798	.6380
x_2		1	-.7071	.7071	x_4		1	.27	-.6478	.1734	-.7418
					x_5			1	-.4666	-.8601	.2064
Variance			1.75	.25	Variance				1.5942	.8507	.555

Moreover, the largest eigenvalue of $\mathbf{R}$ is the same as the largest eigenvalue of $\mathbf{R}_1$, while the second largest eigenvalue of $\mathbf{R}$ is the same as the largest eigenvalue of $\mathbf{R}_2$. Thus, the loadings of the PCs corresponding to the leading eigenvalues of $\mathbf{R}_1$ and $\mathbf{R}_2$ are used to construct the first two (sparse) components of $\mathbf{R}$ with as little as possible loss of information. This remains true if more than two well-defined 'clusters'

of variables are available in the matrix.

The results in this simple example suggest that, if the variables can be grouped into least correlated 'clusters', then approximate sparse principal components can be computed from the correlation matrices of each cluster of variables.

Motivating example 2

In the above simple hypothetical example, the two 'clusters' of variables are uncorrelated and their correlation matrix leads to sparse PCs. Unfortunately, this is not the case for real high-dimensional multivariate data. The correlation coefficient between a pair of variables is hardly ever zero, and each PC contains nonzero-loadings for all original variables. But, the hypothetical example may suggest one thing: to group the variables into clusters in such a way that the correlation between a variable in one cluster and a variable in another cluster is as small as possible.

Now consider a real data set on coal constituents (McCabe, 1984). Table 5.3 contains the correlation matrix of nine constituent elements of coal in 50 samples, together with the loadings of its first four PCs, which account for 85.8% of the total variation.

Table 5.3: *Correlation matrix and PC loadings, coal constituents data*

	Correlation matrix								PC loadings			
Vars	Si	S	Ca	Ti	Fe	Se	Sr	Ba	PC1	PC2	PC3	PC4
Al	.961	.419	-.010	.926	.373	.328	.030	.304	.461	.136	-.319	.133
Si		.454	-.071	.879	.370	.280	-.032	.269	.451	.184	-.291	.138
S			-.058	.425	.657	.465	.061	.225	.356	-.006	.489	-.016
Ca				-.050	.195	.005	.629	.103	.021	-.587	-.097	.560
Ti					.336	.416	.024	.272	.453	.148	-.271	.072
Fe						.424	.093	.185	.323	-.130	.517	.280
Se							.113	.261	.299	-.081	.398	-.302
Sr								.489	.079	-.660	-.178	-.068
BA									.229	-.351	-.185	-.686

The usual interpretation of PCs depends on the magnitude and sign of their loadings. For the coal constituents data, the first PC contains three variables with large (absolute) loadings: Al, Si and Ti; the second PC – two variables {Ca, Sr} with large loadings. Similarly, the third PC has large loadings for {S, Fe, Se}, while the fourth PC has large loading for {Ba}. Thus, the variables might be (subjectively) grouped into four non-overlapping 'clusters': {Al, Si, Ti}, {Ca, Sr}, {S, Fe, Se} and {Ba}. Alternatively, the variables could be grouped into three 'clusters' as {Al, Si, Ti}, {Ca, Sr, Ba}, and {S, Fe, Se} based on the loadings of the first three PCs. However, there is no formal rule for categorizing a loading as small or large. In addition, each PC contains non-zero loadings on all variables. Inevitably, this introduces subjectivity in the PC's interpretation.

For the above 'clusters' of variables, a close look at the correlation matrix reveals that variables in the same group are highly correlated with each other and weakly correlated with the variables from different group. Indeed, the correlation coefficients

corresponding to the first group {Al, Si, Ti} are .961, .926 and .879. These are the three largest correlation coefficients in the matrix. The correlations between each of the variables in this group and the variables in the other groups are relatively small. Similarly, the correlation coefficient between the variables in the second group {Ca, Sr} is .629, while the correlations between the variables in the second group and each of the variables in the other groups are small. The same is true for the third and the fourth groups.

Hence, the absolute sizes of the loadings of the variables in each PC are related to the magnitudes of the correlation coefficients between the variables. Subsets of variables with large correlation coefficients tend to have larger (absolute) loadings in a certain PC than the remaining variables. Thus, the first step in the process of finding interpretable principal components is to cluster the variables. The standard clustering methods may help in this regard. However, we propose a new clustering approach which leads to IPCs which explain as much as possible of the total variance. The resulting IPCs will be later compared with those resulting from the standard clustering methods.

5.3 The weighted-variance clustering method

In this section, a new agglomerative type of clustering method, called weighted-variance, is proposed for clustering variables. The method allows us to either group the variables into a required number k of clusters (like the other existing methods) or choose the 'appropriate' number (and 'best' set) of clusters among all possible sets of clusters.

Let $\mathbf{x}$ denotes a p-vector of variables with correlation matrix $\mathbf{R}$. Our criterion for an optimal clustering of the p variables into k groups involves a function of the variances explained by the linear combinations $z_j = \mathbf{v}_j^\top \mathbf{x}$, $j = 1, 2, \ldots, k$, whose variance is given by $\mathbf{v}_j^\top \mathbf{R} \mathbf{v}_j$. The vector $\mathbf{v}_j$ is found as follows. Assume that the variables are grouped into k non-overlapping clusters and that the jth cluster is composed of q_j variables, $j = 1, \ldots, k$, so that $\sum_{j=1}^{k} q_j = p$. Consider the jth cluster C_j with the first eigenvector $\mathbf{v}_1^{(j)} = (v_{11}^{(j)}, v_{21}^{(j)}, \ldots, v_{q_j 1}^{(j)})$ corresponding to the largest eigenvalue of $\mathbf{R}_j$, the $q_j \times q_j$ correlation matrix of variables in the jth cluster. Let $\boldsymbol{\omega}_j$ be the $q_j \times 1$ vector containing the indices of the original variables clustered into the jth cluster in ascending order, i.e. $\omega_{1,j} \leq \omega_{2,j} \leq \ldots \leq \omega_{q_j,j}$. Define the $p \times q_j$ indicator matrices $\mathbf{G}_j$, for $j = 1, \ldots, k$, as follows: $\mathbf{G}_j$ has 1 at its position $(\omega_{l,j}, l)$ for $l = 1, \ldots, q_j$ and 0 otherwise. Then, put $\mathbf{v}_j = \mathbf{G}_j \mathbf{v}_1^{(j)}$. For the k clusters, let $\mathbf{V}_k$ be a $p \times k$ matrix whose jth column is $\mathbf{v}_j$:

$$\mathbf{V}_k = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k] \, .$$

The aim is to group the variables into k (unknown a priori) clusters such that the sum of variances

$$\tau_k = \sum_{j=1}^{k} \mathbf{v}_j^\top \mathbf{R} \mathbf{v}_j. \tag{5.5}$$

is maximized.

Now, let λ_i $(i = 1, 2, \ldots, p)$ be the ith largest eigenvalue of $\mathbf{R}$, with $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p$. For a reason to be explained in Section 5.4, the eigenvalue λ_j can serve as the weight of the variance of z_j (hence, we call this the weighted-variance clustering method). Here, larger weights are assigned to the variances of the first few linear

combinations. Then, (5.5) can be replaced by the sum of weighted-variances

$$\tau_k = \sum_{j=1}^{k} \lambda_j \mathbf{v}_j^\top \mathbf{R} \mathbf{v}_j. \tag{5.6}$$

If $\mathbf{\Lambda}_k$ denotes a $k \times k$ diagonal matrix whose diagonal elements are given by the λ_j's, then (5.6) can be given in matrix form as

$$\tau_k = \text{trace}(\mathbf{\Lambda}_k \mathbf{V}_k^\top \mathbf{R} \mathbf{V}_k). \tag{5.7}$$

The weighted-variance clustering algorithm starts with p clusters, each containing a single variable, i.e. $C_j = \{x_j\}$, $j = 1, 2, \ldots, p$. That means, there are as many clusters as the number of variables at the first stage. Let τ_p denotes the sum of weighted-variance (5.6) corresponding to the p clusters. We call this stage 0 (no merging takes place). On each subsequent stage, two clusters merge together, reducing the number of clusters by one. At the mth stage ($0 \leq m \leq p-1$), there are $p-m$ clusters available, denoted by $C_j^{(m)}, j = 1, \ldots, p-m$ with $C_j^{(0)} = C_j = \{x_j\}$. At this stage, there are $\begin{pmatrix} p-m \\ 2 \end{pmatrix}$ possible choices each comprising $p-m-1$ 'candidate' clusters for the $(m+1)$th stage. Then, the best choices of clusters at the $(m+1)$th stage are those which, after merging a pair of clusters, maximize

$$\tau_{p-m-1} = \sum_{j=1}^{p-m-1} \lambda_j \mathbf{v}_j^\top \mathbf{R} \mathbf{v}_j, \; m = 0, \ldots, p-2 \tag{5.8}$$

over the $\mathbf{v}_j$'s constructed from all possible 'candidate' clusters obtained at the mth stage. The algorithm on merging a pair of clusters continues either until a required number k of clusters is retained, or all variables are grouped into a single cluster (leading to τ_1). With the latter option, all possible optimal clusters of sizes 1 to p are obtained. This allows us to choose the 'best' clusters from all possible clusters. This

corresponds to a set of $p - m^*$ clusters, say $\mathcal{C}^{(m^*)}$ $(m^* = 0, 1, \ldots, p-1)$, for which τ_{p-m^*} is maximised. Note in this case that we need not fix the number k of clusters a prior, an advantage over the ordinary hierarchical and the k-means methods.

As each pair of components $\mathbf{v}_i^\top x$ and $\mathbf{v}_j^\top x$ $(i < j)$ are correlated to each other, it would be appropriate to replace the variance by adjusted variance (Zou *et al.*, 2006). Let $\mathbf{V}_{p-m} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_{p-m}]$, and let $\mathbf{F}_{p-m}$ be the upper-triangular $(p-m) \times (p-m)$ factor of the Cholesky decomposition of $\mathbf{V}_{p-m}^\top \mathbf{R} \mathbf{V}_{p-m}$, that is

$$\mathbf{V}_{p-m}^\top \mathbf{R} \mathbf{V}_{p-m} = \mathbf{F}_{p-m}^\top \mathbf{F}_{p-m}.$$

As shown by Zou *et al.* (2006), the square of the elements on the main diagonal of $\mathbf{F}_{p-m}$, denoted as $\mathbf{diagF}_{p-m}^2$, gives the vector of adjusted variances. Then, criterion (5.8) is replaced by the adjusted criterion

$$\tau_{p-m-1} = \boldsymbol{\lambda}_{p-m-1}^\top \mathbf{diagF}_{p-m-1}^2, \; m = 0, \ldots, p-2, \tag{5.9}$$

where $\boldsymbol{\lambda}_{p-m-1} = (\lambda_1, \lambda_2, \ldots, \lambda_{p-m-1})^\top$.

The weighted-variance clustering algorithm can be summarized as follows. Denote by $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_k\}$ the set of k $(1 \leq k \leq p)$ clusters, where $\mathcal{C}_j$ is the jth cluster containing q_j variables with $\sum_{j=1}^{k} q_j = p$.

1. Start with p clusters $\mathcal{C}^{(0)} = \{\mathcal{C}_1^{(0)}, \mathcal{C}_2^{(0)}, \cdots, \mathcal{C}_p^{(0)}\}$ where $\mathcal{C}_j^{(0)} = \{x_j\}$, $j = 1, \ldots, p$, is a single-variable cluster.

2. Among the $\begin{pmatrix} p \\ 2 \end{pmatrix}$ possible pairs of clusters from step (1), search for a pair of clusters, say $\mathcal{C}_i^{(0)}$ and $\mathcal{C}_l^{(0)}$, for which the criterion τ_{p-1} in (5.9) has maximum value after merging the two clusters.

3. Merge $C_i^{(0)}$ and $C_l^{(0)}$, and update the set of clusters to $\mathcal{C}^{(1)} = \{C_1^{(1)}, C_2^{(1)}, \cdots, C_{p-1}^{(1)}\}$, which now contains $p-1$ clusters.

4. At the mth stage ($0 \leq m \leq p-2$), search for the pair of clusters, say $C_i^{(m)}$ and $C_l^{(m)}$, among the $\begin{pmatrix} p-m \\ 2 \end{pmatrix}$ possible choices, for which the criterion τ_{p-m-1} in (5.9) has maximum value when the two clusters are merged.

5. Merge $C_i^{(m)}$ and $C_l^{(m)}$, and update the set of clusters to $\mathcal{C}^{(m+1)} = \{C_1^{(m+1)}, C_2^{(m+1)}, \cdots, C_{p-m-1}^{(m+1)}\}$, which now contains $p-m-1$ clusters.

6. Continue merging and updating the clusters until either

   i) a required number k of clusters $\mathcal{C} = \mathcal{C}^{(p-k)}$ is reached, or

   ii) $m = p-2$, leading to a single cluster $\mathcal{C}^{(p)}$ containing all the variables. In this case, the best set of clusters, say $\mathcal{C} = \mathcal{C}^{(m^*)}$, is chosen to be the one for which τ_{p-m^*} $(0 \leq m^* \leq p-1)$ is maximum.

5.4 Interpretable principal components

Now, assume that the variables are already grouped into k clusters using an arbitrary clustering technique and let q_j denote the number of variables in the jth cluster with $\sum_{j=1}^{k} q_j = p$. The next step is to construct the IPCs.

5.4.1 Constructing IPCs

The nonzero loadings of the jth IPC are obtained from the eigenvector corresponding to the largest eigenvalue of the correlation matrix of the variables in the jth cluster.

Let $\mathbf{x}_j$ be the vector of q_j variables in the jth cluster with correlation matrix $\mathbf{R}_j$, and let $\mathbf{v}_1^{(j)}$ be the eigenvector corresponding to the largest eigenvalue of $\mathbf{R}_j$. Then, the p-vector $\mathbf{v}_j$ $(j = 1, 2, \ldots, k)$ is formed from $\mathbf{v}_1^{(j)}$ in such a way that the q_j loadings of $\mathbf{v}_1^{(j)}$ become the nonzero-loadings of $\mathbf{v}_j$ for the same set of variables $\mathbf{x}_j$, while the remaining $(p-q_j)$ loadings of $\mathbf{v}_j$ are zeros. The vector $\mathbf{v}_j$ is called the jth interpretable principal component (IPC). In particular, if the weighted-variance clustering procedure of Section 5.3 is employed, the IPCs are available as by-products.

The expression $\mathbf{v}_j^\top \mathbf{R} \mathbf{v}_j$ in (5.8) gives the variance accounted for by the jth IPC. From the property of ordinary PCA, principal components (PCs) are presented in a decreasing order of their variances so that the first few PCs explain the majority of the variation in the original data. We also order the IPC's in the decreasing order of their variances. To allow the IPCs keep this property, we use the variances of the first k PCs (λ_j's) as weights attached to the variances of the corresponding ordered IPCs. Thus, the idea behind incorporating weights in (5.8) is to identify those k IPCs which preserve as much as possible the explanatory power of the first k PCs. In general, the criterion to be maximized is simply the sum of the weighted-variance of the IPCs, where λ_j serves as the weight for the variance of the jth IPC.

If the clustering of variables is 'optimal' with respect to the maximal value of the criterion (5.9), the performance of the IPCs can be assessed using the cumulative percentage of variance explained by the components. However, as the IPCs are correlated with each other, the cumulative percentage of adjusted variance (Zou *et al.*, 2006) is a better measure of goodness-of-fit.

5.4.2 Number of IPCs

In PCA, there is no hard and fast rule for deciding the number k of PCs to retain in the process of reducing dimensionality. Some of the *ad hoc* rules-of-thumb used in practice include the cumulative percentage of total variation and the scree plot (Jolliffe, 2002, Section 6.1). The former rule suggests retaining the first k ($< p$) PCs which explain a required cumulative percentage of total variation (say 80% or 90%) in the original data, while in the latter rule, one selects the value of k from a scree plot. However, the choice of k is subjective in both cases. Another rule suggests to exclude those PCs whose eigenvalues are less than the average, $[\sum_{i=1}^{p} \lambda_i]/p$. For correlation PCs, the numerator is equal to p and hence the average is 1.

On the other hand, the number of IPCs depends on the number of clusters. However, there is no simple rule for choosing the number of clusters in cluster analysis, though there are some suggestions (Seber, 2004, p.388). For the hierarchical linkage clustering method, the required number of clusters can be inferred from the nodes of the dendrogram. In an extreme case, all variables fall into one cluster, in which case the single sparse component is the same as the first PC. In another extreme case, each variable forms a cluster, resulting in a total of p sparse components. Thus, the number of sparse components ranges from 1 to p. However, neither of the two extreme cases is interesting as the objective is to obtain interpretable components in a reduced dimension, which recover as much as possible of the total variation in the data. In general, the investigator may (subjectively) decide on the number of components to work with, depending on the trade-off between the required level of sparsity, the dimensionality and the cumulative percentage of explained variance.

The weighted-variance clustering algorithm proposed in Section 5.3 runs under two options – either until a required number k of clusters is obtained, or until all original variables are grouped into a single cluster. The first option involves subjective judgement like the standard clustering methods, but the second option helps to obtain all possible clusters of variables without fixing k *a priori*. Then, the 'best' solution is the number of clusters in the configuration which give the maximum value of criterion (5.9). This is equivalent to identifying the value of k in (5.9), which gives the maximum value of τ_k:

$$\tau_{opt} = \max \tau_k, \ k = 1, \ldots, p.$$

Here, the values of τ_k can be plotted against k to give the cluster graph. Generally in practice, such a graph has the shape of a downward-facing parabola, in that it increases to a maximum and then decreases thereafter. Then, the 'best' value of k corresponds to the peak of the graph. This graph may also be used as an alternative tool for deciding the number of PCs to retain in the ordinary PCA.

The level of sparsity of a component is also affected by the number of components. Unlike the constrained sparse techniques (such as the LASSO-based methods), which control the number of nonzero loadings per sparse component by introducing a tuning parameter, the IPC approach regulates the level of sparsity via the number of clusters. The higher the number of clusters the sparser the components, due to the property that the variables are non-overlapping in each sparse component. The feature of the IPC method not being dependent on a tuning parameter adds one more advantage over other similar methods.

5.4.3 Principal components, clusters and variable selection

The idea behind some of the variable selection methods based on PCA is to reduce the number of variables without sacrificing too much information about the original data set. One of the variable discarding methods proposed in Jolliffe (1972) is the principal components method, which associates one variable with each PC for discarding or retaining purposes. This method, in general, performs PCA and associates one variable to each of the last $(p - k)$ components, namely the variable which has the largest coefficient in the component. Some criteria are proposed to choose the last $(p - k)$ components. Then the variables associated with these components are rejected. Another method associates one variable with each of the first k components for retaining k variables.

A potential relationship between cluster analysis and variable selection is that one variable could be retained from each cluster as representative of the cluster. Jolliffe (1972) discusses the use of cluster analysis as a variable discarding technique. The idea is that if the p original variables are grouped into k clusters based on a certain optimality criterion, then each cluster can be represented by a single variable from the cluster and the remaining $(p - k)$ variables discarded. He considers two of the agglomerative hierarchical clustering methods for this purpose – the single-linkage and the average-linkage clustering methods. He also discusses techniques of selecting a representative variable from the group of variables in a cluster. Jolliffe (1973) applied these techniques to real data sets. Similarly, McCabe (1984) proposed a number of criteria for identifying 'principal variables', where essentially the idea is that a single representative variable can replace a cluster of variables.

Clustering techniques have also been used as a companion to PCA. Vigneau and Qannari (2003) proposed a clustering method in which correlated variables lump together based on a criterion involving the squared correlation between a variable in a cluster and the leading principal component of the covariance matrix of the cluster. In addition, the 'gene shaving' algorithm (Hastie *et al.*, 2000), which deals with clustering of genes with similar expression, involves discarding ('shaving') a given proportion of genes having the smallest absolute correlation with the leading principal component.

A common problem with using clustering methods for variable selection is that the number of representative variables depends on the number k of clusters, which is often decided subjectively. To overcome this, Jolliffe (1972) relates the required number of clusters to some threshold r_0 such that the amalgamation of the clusters continues until the value of the clustering criterion first falls below r_0. Then, the number k of clusters formed at this stage is the required solution. However, there is still no hard and fast rule for finding the value of r_0.

One advantage of the weighted-variance clustering algorithm over the standard hierarchical algorithms could be that it chooses the 'best' clusters without fixing k or r_0 in advance. That means, having decided to use (5.9), no further choices need be made. It can also serve as an alternative hierarchical clustering method to obtain a required number k of clusters.

Example (continued)

Consider again the correlation matrix of the coal constituents data in Table 5.3. Let the variables Al, Si, S, Ca, Ti, Fe, Se, Sr and Ba be denoted by the serial numbers from 1 to 9. The left-hand plot in Figure 5.1 gives the dendrogram of the clustered vari-

ables based on the average-linkage method (see Chapter 2 for the standard clustering methods). With a required number of three clusters, the weighted-variance clustering method results in the following clusters: {1,2,3,5,6,7}, {4,8}, and {9}, which are identical to the ones given by the dendrogram in Figure 5.1. However, the k-means method with $k = 3$ results in clusters {1,2,5}, {3,6,7}, and {4,8,9}. If four clusters are required, then each of the three approaches results in the same set of clusters: {1,2,5}, {4,8}, {3,6,7}, and {9}.

If the weighted-variance clustering algorithm is allowed to run without fixing the number of clusters a priori, then the value of criterion (5.9) is maximum when the variables are divided into three clusters {1,2,3,5,6,7}, {4,8}, and {9}. This 'best' number of clusters is also shown by the cluster plot in Figure 5.1, which relates the number of clusters with the maximum value of the criterion (5.9). As noted above, the same clusters are found from the dendrogram if three clusters are sought. However, the dendrogram does not clearly indicate the 'best' number of clusters, as the dissimilarity drop may equally suggest either 3 or 5 clusters.

McCabe (1982) identified a few possible four-variable subsets of principal variables for the coal constituents data. The subset with the largest percentage of variation explained is found to be {2, 4, 6, 9}. Note that each of these principal variables correspond uniquely to one cluster in the four-clusters case. For the case of three-clusters, he identified four sets of principal variables with large percentage of variation explained, of which three sets fulfil the property that each of the principal variables in a set correspond uniquely to one cluster.

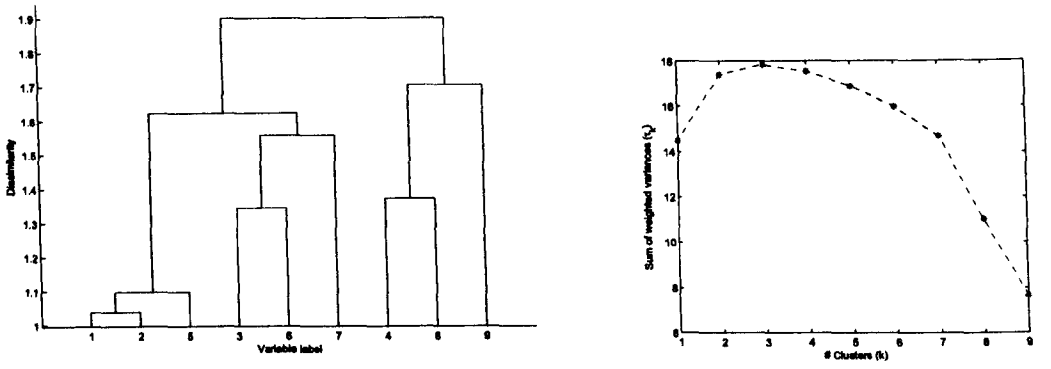

Figure 5.1: *Dendrogram (Left) and cluster plot (Right) for the coal constituents data.*

The first three PCs of the correlation matrix account for 76.1% of the total variation while the first four PCs account for 85.8%. If we decide to work with four clusters, then the nonzero-loading variables for the corresponding IPCs become $\{1, 2, 5\}$, $\{4, 8\}$, $\{3, 6, 7\}$, and $\{9\}$. The variables in each cluster correspond to the large-loading variables of the PCs in Table 5.3. The corresponding IPCs account for 76.1% of the total adjusted variance. On the other hand, the three IPCs based on the weighted-variance clustering method contain nonzero-loadings for sets of variables $\{1, 2, 3, 5, 6, 7\}$, $\{4, 8\}$, and $\{9\}$, respectively, and account for 67.4% of the total adjusted variance.

5.5 Further Applications

In this section, two synthetic and two real data sets are considered to illustrate the IPCs method. The new weighted-variance and two other standard clustering methods are employed for clustering variables, with more attention to the weighted-variance clustering. The corresponding IPCs are computed using the methods given in Section 5.4.

Synthetic data I

We use a simple example to see if the weighted-variance clustering method is able to reconstruct the groups of variables given by the data-generating model. It might also help to see if the corresponding IPCs can reveal the important features of the PCs without sacrificing much information. For this purpose, consider again the artificial data generated for ten variables as given in Table 4.5 (Section 4.3 of chapter 4), which is based on one of the models considered in Jolliffe (1972).

By construction, the 10 variables x_i fall into 4 groups: $\{x_1\}$, $\{x_2, x_3\}$, $\{x_4, x_5, x_6\}$ and $\{x_7, x_8, x_9, x_{10}\}$. The variables forming each group are linear combinations of the variables within the same group plus random disturbances, whereas variables from different groups are independent.

The correlation matrix $\mathbf{R}_{10}$ of the ten variables x_i is computed based on 100 random observations. The main results were stable over different random samples of the same size. The correlation coefficient between a variable from one group and a variable from another group is very small, while the correlation coefficients ρ_{ij} between the variables i and j from the same group are found to be as follows: $r_{23} = .800, r_{45} = .865, r_{46} = .795, r_{56} = .804, r_{78} = .925, r_{79} = .926, r_{7,10} = .930, r_{89} = .905, r_{8,10} = .933, r_{9,10} = .954$.

First, the weighted-variance clustering method is applied to $\mathbf{R}_{10}$ with a required number of four clusters. The loadings and the variances of the corresponding IPCs, together with that of the PCs, are given in Table 5.4. The four IPCs perfectly identify the important features of the first four PCs and explain nearly the same cumulative variance.

Table 5.4: *Loadings and cumulative variance (CV) of the PCs and IPCs, synthetic data 1. Empty cells have zero values.*

	PC loadings				IPC loadings			
Variable	PC1	PC2	PC3	PC4	IPC1	IPC2	IPC3	IPC4
x_1	.004	-.047	-.119	.985				1
x_2	.116	.129	-.672	-.155			-.707	
x_3	.093	.102	-.696	.001			-.707	
x_4	-.040	.572	.084	.032		.582		
x_5	-.039	.570	.125	.049		.584		
x_6	-.003	.563	.053	.035		.567		
x_7	.491	.010	.105	.021	.499			
x_8	.492	-.001	.064	-.024	.497			
x_9	.494	.034	.069	.010	.500			
x_{10}	.498	.001	.069	.028	.504			
CV(%)	38.4	65.1	82.6	92.5	37.9	64.3	82.0	91.9

Next, the weighted-variance clustering method is applied without fixing the number of clusters a priori. The cluster plot in Figure 5.2 relates the number of possible clusters with the value of the criterion (5.9). The plot shows that the criterion is maximized when the variables are grouped into four clusters. Moreover, each cluster is found to contain the same set of variables as required by the model.

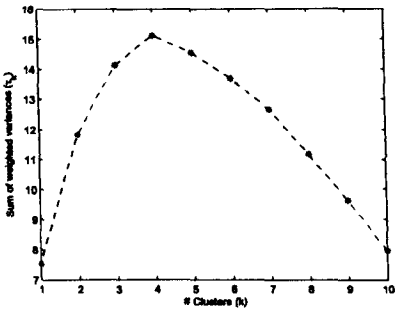

Figure 5.2: *Cluster plot for synthetic data 1.*

Synthetic data II

Consider a synthetic data set generated as follows (Zou *et al.*, 2006):

$$V1 \sim N(0, 290), \quad V2 \sim N(0, 300),$$

$$V3 = -0.3V1 + 0.925V2 + \epsilon, \quad \epsilon \sim N(0, 1),$$

and $V1$, $V2$, and ϵ are independent normal variates. Then 10 observable variables are constructed as follows:

$$\begin{aligned} X_i &= V_1 + \epsilon_i^1, \quad \epsilon_i^1 \sim N(0,1), \quad i = 1,2,3,4, \\ X_i &= V_2 + \epsilon_i^2, \quad \epsilon_i^2 \sim N(0,1), \quad i = 5,6,7,8, \\ X_i &= V_3 + \epsilon_i^3, \quad \epsilon_i^3 \sim N(0,1), \quad i = 9,10, \end{aligned}$$

where $\{\epsilon_i^j\}$ are independent, $j = 1,2,3$; $i = 1,2,\ldots,10$.

We generate 1000 random observations for each of the ten variables and the weighted-variance clustering method is applied to the (10×10) matrix of correlations. The criterion (5.9) is maximum when the data are grouped into two clusters: $\{X_1, X_2, X_3, X_4\}$ and $\{X_5, X_6, X_7, X_8, X_9, X_{10}\}$. The corresponding IPCs, together with the results of

PCA, simple thresholding (ST), and SPCA (Zou *et al.*, 2006) are given in Table 5.5. The second sparse component for each of the sparse methods is the same, but the first sparse component differs. The variables X_9 and X_{10} are included in the nonzero-loading variables of the first IPC. This is due to the fact that the weighted-variance clustering method allocates each variable into one of the two clusters, and V_3 is highly correlated to V_2, but weakly to V_1. The first ST component also includes these two variables, but excludes X_5 and X_6. Thus, the first IPC fits the first PC much better than the first components from both SPCA and ST.

Table 5.5: *Loadings and cumulative variance (CV) of components from PCA, SPCA, ST, and IPC methods for synthetic data 2. The empty cells are 0s.*

	PCA		SPCA ($\lambda = 0$)		ST		IPC	
Variable	1	2	1	2	1	2	1	2
x_1	-.116	.478		.5		.5		.5
x_2	-.116	.467		.5		.5		.5
x_3	-.116	.478		.5		.5		.5
x_4	-.116	.478		.5		.5		.5
x_5	.395	.146	.5				.4	
x_6	.395	.146	.5				.4	
x_7	.395	.146	.5		.5		.4	
x_8	.395	.146	.5		.5		.4	
x_9	.401	-.010			.5		.4	
x_{10}	.401	-.010			.5		.4	
CV(%)	60.0	99.6	40.9	80.4	38.8	77.4	58.9	98.1

The 1988 Olympic decathlon data

This data contain results of the 1988 Olympic decathlon for 33 competitors (Everitt and Dunn, 2001, pp. 20 and 57). The ten events (variables) are 100m (x_1), long jump

(x_2), shot putt (x_3), high jump (x_4), 400m (x_5), 110m hurdles (x_6), discus (x_7), pole vault (x_8), javelin (x_9) and 1500m (x_{10}). We consider the correlation matrix of the ten events, as reproduced in Table 5.6.

Table 5.6: *Correlation matrix of events for the 1988 Olympic decathlon (Everitt and Dunn, 2001)*

Events	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9
x_2	0.540								
x_3	0.208	0.142							
x_4	0.146	0.273	0.122						
x_5	0.606	0.515	-0.095	0.088					
x_6	0.638	0.478	0.296	0.307	0.546				
x_7	0.047	0.042	0.806	0.147	-0.142	0.110			
x_8	0.389	0.350	0.480	0.213	0.319	0.522	0.344		
x_9	0.065	0.182	0.598	0.116	-0.120	0.063	0.443	0.274	
x_{10}	0.261	0.396	-0.269	0.114	0.587	0.143	-0.402	0.031	-0.096

Application of the weighted-variance clustering approach to the 1988 Olympic decathlon data results in three 'best' clusters – $\{x_1, x_2, x_5, x_6, x_8, x_{10}\}$, $\{x_3, x_7, x_9\}$ and $\{x_4\}$. The 'best' number of clusters is indicated by the peak of the cluster plot in Figure 5.3. For a required number of three clusters, the dendrogram of the hierarchical linkage method given on the left-hand plot in Figure 5.3 suggests the same set of clusters. On the other hand, the k-means method with $k = 3$ groups the events into three as $\{x_1, x_2, x_5, x_6, x_{10}\}$, $\{x_3, x_7, x_8, x_9\}$ and $\{x_4\}$. The difference might be attributed to the fact that the two clustering methods are based on different criteria. Unlike the k-means, the target of the weighted-variance method is to approximate PCs, which is beyond a mere clustering.

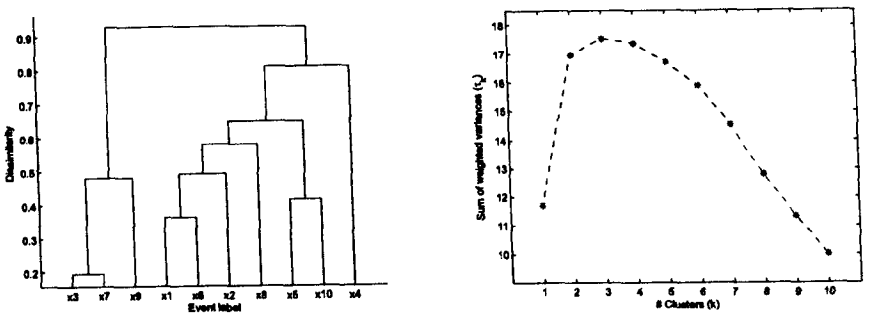

Figure 5.3: *Average-linkage dendrogram (Left) and cluster plot (Right) for the 1988 Olympic decathlon data.*

PCA of the decathlon data shows that only the first two PCs have variances greater than 1, while the classical scree plot (not shown here) suggests consideration of the first three PCs which accounts for 69.7% of the total variation. This result is similar to the one suggested by our cluster plot. But, note that using the peak of a cluster is more objective than using the elbow of a classical scree plot for determining the number of clusters.

The loadings of the first PC are all positive, giving the usual interpretation of overall performance. However, the first IPC, based on the weighted-variance clustering method, contains positive loadings only for the six variables in the first cluster while the remaining loadings are zeros. This relates the first IPC to the performance of the running events. The second IPC has nonzero-loadings for the three 'power' events – shot, discus and javelin. The third IPC is composed only of the high-jump event. The IPC loadings and explained variances are slightly different to those based on the k-means clustering method. The k-means method tends to produce clusters of similar sizes, which might not lead to IPCs with desirable properties. The PCs and IPCs

together with the cumulative variance are given in table 5.7.

Table 5.7: *Component loadings and cumulative adjusted variance (CAV) using PCA, IPC based on k-means (KM), and IPC based on weighted variance (WV) for the correlation matrix of the 1988 Olympic decathlon. Empty cells have zero values.*

	PC loadings			IPC loadings (KM)			IPC loadings (WV)		
Events	PC1	PC2	PC3	IPC1	IPC2	IPC3	IPC1	IPC2	IPC3
x_1	.42	-.15	-.27	.48			.46		
x_2	.39	-.15	.17	.45			.43		
x_3	.27	.48	-.10		.59			.63	
x_4	.21	.03	.85			1.00			1.00
x_5	.36	-.35	-.19	.50			.47		
x_6	.43	-.07	-.13	.44			.44		
x_7	.18	.50	-.05		.54			.59	
x_8	.38	.15	-.14		.39		.33		
x_9	.18	.37	.19		.46			.51	
x_{10}	.17	.42	.22	.35			.29		
CAV(%)	34.2	60.2	69.7	29.2	54.0	63.2	31.8	54.0	63.2

The Pitprop data

As considered in Chapter 4, the pitprop data (Jeffers, 1967) contains 13 variables $x_1, x_2, \ldots, x_{13}$ measured on 180 pitprops cut from Corsican pine timber. Jeffers (1967) considered the first six PCs of the correlation matrix for further analysis and interpretation. Their cumulative percentages of variance explained are 32.4%, 50.7%, 65.1%, 73.6%, 80.6% and 86.9%.

We consider four clustering solutions – one based on a particular dendrogram, one based on k-means, and two based on the weighted-variance method (with and without

fixing the number of components a priori). The left-hand plot in Figure 5.4 gives the dendrogram based on the average-linkage method. For a required number of six clusters (chosen for this data set for a reason given in Chapter 4, Section 4.1.2), the dendrogram groups the variables as $\{x_1, x_2, x_6, x_7, x_8, x_9, x_{10}\}$, $\{x_3, x_4\}$, $\{x_5\}$, $\{x_{11}\}$, $\{x_{12}\}$ and $\{x_{13}\}$, while the k-means method groups them as $\{x_1, x_2, x_8, x_9, x_{10}\}$, $\{x_3, x_4\}$, $\{x_6, x_7\}$, $\{x_{12}, x_{13}\}$, $\{x_5\}$ and $\{x_{11}\}$. For the same required number of clusters, the weighted-variance clustering approach suggests the same set of clusters as the dendrogram. On the other hand, if the weighted-variance algorithm is allowed to run without fixing the number of clusters, then the 'best' number of clusters is found to be six (see the cluster plot in Figure 5.4). The corresponding clusters are again the same as the ones derived from the dendrogram.

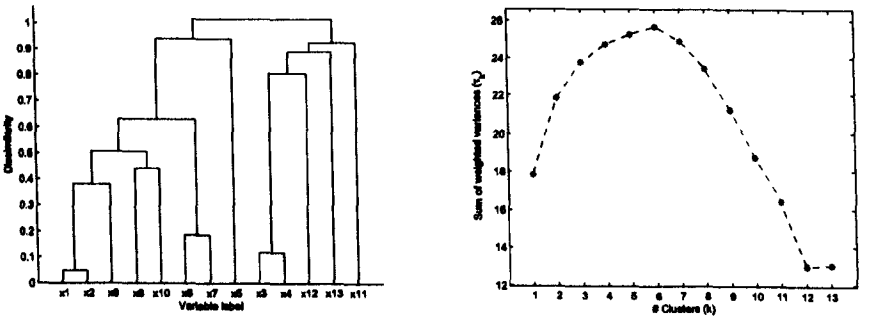

Figure 5.4: *Average-linkage dendrogram (Left) and cluster plot (Right) for the Pitprop data.*

Now we construct the corresponding IPCs. If the variables are grouped into six clusters (based on either the dendrogram or the weighted-variance), then the corresponding six IPCs explain 76.0% of the total variance, while the cumulative adjusted variance is 73.5%. The number of nonzero loadings (which measures the sparsity level) in the six IPCs are 7, 2, 1, 1, 1 and 1. The cumulative adjusted variance explained by

the six IPCs based on the k-means method is 71.1%.

The Pitprop data set has become a benchmark example and is used in nearly every paper studying sparse PCs. In the remaining part of this illustration, the IPCs based on the weighted-variance method with six clusters are compared with the sparse components obtained by other methods. As most of the methods produce 4th, 5th and 6th sparse components with a single nonzero (unit) loading, Table 5.8 contains the loadings of the first three sparse components only, and the corresponding cumulative variance (CV) and adjusted variance (CAV). This table is part of the table given in Chapter 4. The values in the table are collected from the original papers. The abbreviations are: SPCA – sparse principal component analysis (Zou *et al.*, 2006), SCoTLASS – simplified component technique-LASSO (Jolliffe *et al.*, 2003) with $\tau = 1.75$, DSPCA – direct sparse PCA (d'Aspremont *et al.*, 2007), ESPCA – exact sparse PCA (Moghaddam *et al.*, 2006), IDR – interpretable dimension reduction (Chipman and Gu, 2005) with sparsity constraint $\eta = .9$, and sBarse – sparse biplots component analysis (Chapter 4, this thesis).

Table 5.8: *Sparse loadings and variance of the first three components explained by different methods, Pitprop data. Empty cells have zero values, while* 0* *indicates zero to 2 decimal places.*

Method	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	CV	CAV	0s
SPCA 1	.48	.48			-.18		.25	.34	.42	.40				28	28	6
SPCA 2			.79	.62				-.02				.01		42	42	9
SPCA 3					.64	.59	.49						-.02	57	55	9
SCoTLASS 1	.66	.68					0*	0*	.28	.11				20	20	7
SCoTLASS 2		0*	.64	.70		.29	.11					0*		36	33	7
SCoTLASS 3			.20	0*		-.17	-.66			0*			.70	50	46	7
DSPCA 1	.56	.58					.26	.10	.37	.36				27	27	7
DSPCA 2			.71	.71										42	40	11
DSPCA 3						-.79	-.61						.01	56	50	10
ESPCA 1	.48	.49					.41		.42	.43				26	26	8
ESPCA 2			.71	.71										41	40	11
ESPCA 3						-.81	-.58							55	49	11
IDR S1	-.42	-.42				-.30	-.42	-.31	-.37	-.39				31	31	6
IDR S2			-.69	-.58								-.44		45	45	10
IDR S3				.43	.58	.57							-.39	59	56	9
IPC 1	.42	.43				.27	.40	.31	.38	.40				31	31	6
IPC 2			.71	.71										45	45	11
IPC 3					1									53	52	12
sBarse 1	-.41	-.41					-.41	-.41	-.41	-.41				29	29	7
sBarse 2			.71	.71										43	43	11
sBarse 3					.71	.71								54	53	11

The first two components of IPC outperform that of the SPCA with respect to both the sparsity level and the cumulative percentage of explained variance. With the same number of nonzero loadings (same sparsity level), the first IPC explains a higher percentage of variance than the first component of SPCA. In addition, the second IPC, accounting for 14% of the total adjusted variance, contains only two nonzero-loading variables. But, the second SPCA component, accounting for the same percentage of total adjusted variance as the second IPC, has four nonzero-loading variables (and hence less sparse). Considering the first three components, IPC performs better than

SPCA with respect to the level of sparsity, but not with respect to the explained cumulative adjusted variances.

Similarly, the IPC method performed better than SCoTLASS, when considering the first two or the first three sparse components. In addition, the first three components from IPC are sparser and explain a higher cumulative percentage of adjusted variance than the corresponding components from DSPCA. Compared to ESPCA, the first three IPCs account for a larger cumulative percentage of adjusted variance for the price of one nonzero-loading variable. The IDR components, accounting for higher cumulative percentage, are quite sparse, but they lack orthogonality. Finally, comparison of the IPC method with the sBarse method (Chapter 4) shows that the first two sparse components of the former explain higher cumulative percentage of variance (45%) than the corresponding components of the latter (43%), while the number of zero-loading variables are the same in both cases (which is 11). Considering the first three sparse components, however, the sBarse method performs better than the IPC method with respect to the explained cumulative percentage of variance, but vice versa with respect to the level of sparsity measured by the number of zero-loading variable.

Sparse principal components with non-overlapping variables are expected to give simpler and possibly clearer interpretation than those with overlapping variables. The IPCs are designed in such a way that a variable gets a nonzero loading in only one sparse component. This property is not common for the other methods, e.g. DSPCA, ESPCA and IDR.

The IPCs results can be compared with that of variable selection. The IPC with only one nonzero loading might indicate that this particular variable could be one of the original variables to be retained in the variable selection process. For example,

the last four (of six) IPCs of the Pitprop data set are composed of a single variable, namely, the 5^{th}, 11^{th}, 12^{th} and 13^{th}, respectively. On the other hand, the variable selection technique proposed by Jolliffe (1973) identified the variables x_1, x_3, x_5, x_{11}, x_{12} and x_{13}. But, the method by Cadima and Jolliffe (2001) contain x_2 instead of x_1, i.e., the selected variables are $\{x_2, x_3, x_5, x_{11}, x_{12}, x_{13}\}$. McCabe (1984) also found the latter subset of variables as the ones explaining the largest percentage of variation among the possible subsets of six principal variables. Clearly, the IPCs for the Pitprop data are in agreement with the results from these three variable selection methods.

5.6 Summary

This chapter is motivated by the specific form of the eigenvalue decomposition of a block-diagonal correlation matrix, where PCA of such a matrix results in a sparse loadings matrix. A clustering approach is proposed for approximating a real data matrix by a block-diagonal matrix, so that sparse principal components can be constructed from the data or correlation matrix of each cluster of variables. For this purpose, a weighted-variance clustering approach is proposed, which can be applied to data sets with smaller number of variables than observations.

Different types of data sets are used for illustrating the technique, and the resulting cluster-based sparse PCs are compared with those based on existing methods. The results show that the sparse PCs based on the weighted-variance clustering method perform well with respect to their percentage of cumulative adjusted variance explained and their level of sparsity.

Chapter 6

Sparse principal components by semi-partition clustering

In Chapter 5, we proposed a clustering approach to interpretable principal components (IPCs) in which the variables are first grouped into clusters, and then the IPCs are computed from the correlation or data matrix of each cluster, leading to sparse components with non-overlapping variables. However, due to the design, the clustering algorithm can only be applied to those data sets with a smaller number of variables (p) than observations(n).

In this chapter, we propose a sparse principal components method based on clustering which can be applied to data sets with either $n > p$ or $p \gg n$. The ultimate objective of the chapter is to construct cluster-based sparse principal components (CSPCs) from the data or correlation matrix of each cluster, which share some of the basic properties of the standard PCs. One such property is variance maximization. Thus, we search for a small number of clusters of variables such that the cumulative adjusted variance (Zou *et al.*, 2006) explained by the corresponding CSPCs is max-

imized. However, existing standard clustering methods are not designed in this way and, thus, may not lead to clusters with such properties. As a result, we propose a new type of clustering approach, called the semi-partition, which is assumed to give the intended types of clusters. Here, note that we are not intending to propose a "better" clustering technique than the existing ones; rather, we are proposing a clustering approach which leads to "better" CSPCs.

Examples on small as well as large data sets are considered, but more attention is given to microarray gene expression data sets. Such data sets are characterized by having tens and hundreds of thousands of genes (considered here as variables) while the number of samples rarely exceeds a hundred. The information contained in the data matrix is often overshadowed by the size of the data, and clustering is often used to uncover the information. The purpose of gene-clustering in gene expression data analysis might be to find genes that are potentially co-expressed, which has significant biological importance. For instance, gene-shaving (Hastie *et al.*, 2000) is such a method aimed to identify small subsets of genes with coherent expression patterns and large variation across samples.

The chapter is organized as follows. The semi-partition clustering approach is proposed in Section 6.1 and the cluster-based sparse component method is outlined in Section 6.2. In Section 6.3, the developed technique is applied to two simple data sets with $n > p$ (one synthetic and another real) and to two gene expression data sets with $p \gg n$. A short summary of the chapter is given in Section 6.4.

6.1 The semi-partition clustering approach

The semi-partition clustering algorithm forms clusters of variables sequentially in two steps. First, the elements of a vector of variables $\mathbf{x}$ are ordered (sorted) based on one criterion, and then the ordered variables are partitioned based on another criterion. At each stage of cluster formation, a type of *semi-partition* clustering is used in which the ordered vector of variables are partitioned into two groups, say $[\mathbf{x}_1|\mathbf{x}_2]$. The partitioning is made at the position of the "weakest-link" in the ordered variables where the 'gap' between the groups is maximal or the link is weak. Then, the first subgroup ($\mathbf{x}_1$) forms the first cluster, while the other subgroup is subject to new ordering and partitioning. In other words, only one of the two subgroups at a specific stage is subject to further partitioning at the next stage (and hence the name semi-partition). Unlike the standard partition clustering method, which simultaneously assigns each variable into one of the k clusters (a number fixed *a priori*), the semi-partition method forms clusters in a recursive way. Thus, the procedure continues until one of the two options is satisfied – either a required number k of clusters is obtained or no more ordering and/or partitioning procedure is feasible (see Section 6.1.3).

As already pointed out, the proposed method can be applied to either small or large data sets, but this section is developed based on a gene expression data set as the main target so that clustering is made to the genes.

6.1.1 Gene-ordering

A gene expression data set with p genes and n samples is usually expressed as a $p \times n$ matrix $\mathbf{W}$. But from now on, we work with the $n \times p$ matrix $\mathbf{X} := \mathbf{W}^\top$ in this chapter.

Let $\mathbf{R}$ denotes the matrix of correlations for the p genes. Suppose that the highest correlation coefficient is r_{ij}, the correlation coefficient between the ith and the jth genes. Then form a set of two indices, say $\mathbf{s}^{(2)}$, with elements $s_1^{(2)} = i$ and $s_2^{(2)} = j$. Now, choose a third gene which is highly correlated to both the ith and the jth genes. That is, identify a gene with index k, that maximizes $r_{ik} + r_{jk}$ $(i \neq j \neq k)$ and set $s_3^{(3)} = k$, so that $\mathbf{s}^{(3)} = [\mathbf{s}^{(2)}|s_3^{(3)}]$. Then, select the fourth gene, say $s_4^{(4)} = l$, which maximizes $r_{il} + r_{jl} + r_{kl}$ and forms $\mathbf{s}^{(4)} = [\mathbf{s}^{(3)}|s_4^{(4)}]$, and so on. The procedure continues in a similar way until all the genes have been ordered. In general, the $(q+1)$th ordered gene, say m, will be the one that maximizes

$$f_1(\mathbf{s}^{(q)}, m) = \sum_{i=1}^{q} r_{s_i^{(q)},m} \text{ , over all } m \notin \mathbf{s}^{(q)} \text{ , } q = 2, 3, \ldots \tag{6.1}$$

An alternative criterion might be to take the sum of the squares of the correlation coefficients. That is, replacing $r_{s_i^{(q)},m}$ in (6.1) by $r^2_{s_i^{(q)},m}$. However, this option is not considered here as it ignores the signs of the correlation coefficients, which may have a significant effect on the final result. This and other similar options, such as the use of $|r_{s_i^{(q)},m}|$ will be studied elsewhere.

A similar measure can be developed if the correlation coefficient $r_{s_i^{(q)},m}$ in (6.1) is replaced by a distance (dissimilarity) measure $(1 - r_{s_i^{(q)},m})$, but this time, the $(q+1)$th gene will be the one that minimizes the sum of distances between the gene and each of the q genes. The distance measure has similar features with the inter-cluster dissimilarity measure used in the average linkage hierarchical clustering method, if the q ordered genes are considered as one cluster and the single $(q+1)$th gene as another cluster (Everitt and Dunn, 2001, Section 6.2).

The vector of ordered genes is used in Section 6.1.2 to form a cluster. Then, the

ordering algorithm repeats on the remaining unclustered genes, and so on.

6.1.2 Gene-partitioning

Once the vector of indices of the ordered genes $\mathbf{s} \equiv \mathbf{s}^{(p)}$ is found, we require a partitioning criterion. Consider the p-vector of ordered genes $\mathbf{x}$ corresponding to $\mathbf{s}$, and re-arrange the correlation matrix $\mathbf{R}$ accordingly. [For the sake of simplicity, the formulation of the criterion is based on the correlation matrix of $\mathbf{x}$, though the computations later involve the data matrix.] It is known that the variance of a linear combination $y = \mathbf{a}^\top \mathbf{x}$, given by $\mathbf{a}^\top \mathbf{R}\mathbf{a}$, is maximized if $\mathbf{a}$ is the eigenvector corresponding to the leading eigenvalue of $\mathbf{R}$ (constrained to $\mathbf{a}^\top \mathbf{a} = 1$). Now, let $\mathbf{s}$ be partitioned into two vectors as $\mathbf{s} \equiv [\mathbf{s}_1 \mid \mathbf{s}_2]$ having k_1 and $p - k_1$ genes. Then, $\mathbf{R}$ can be rewritten as the following block-matrix:

$$\mathbf{R} = \begin{bmatrix} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \end{bmatrix},$$

where $\mathbf{R}_{ii}$ (i=1,2) is the correlation matrix of the vector of genes $\mathbf{x}_i$ corresponding to $\mathbf{s}_i$, and each element of the matrix $\mathbf{R}_{12} = \mathbf{R}_{21}^\top$ refers to the correlation coefficient between a gene from $\mathbf{x}_1$ and a gene from $\mathbf{x}_2$. If $\mathbf{a}_1$ and $\mathbf{a}_1'$ denote the eigenvectors corresponding to the leading eigenvalues of $\mathbf{R}_{11}$ and $\mathbf{R}_{22}$, respectively, then the matrix consisting of the variances and covariances of $y_1 = \mathbf{a}_1^\top \mathbf{x}_1$ and $y_1' = \mathbf{a}_1'^\top \mathbf{x}_2$ is

$$\mathbf{C}_{y_1 y_1'} = \begin{pmatrix} \mathbf{a}_1^\top \mathbf{R}_{11} \mathbf{a}_1 & \mathbf{a}_1^\top \mathbf{R}_{12} \mathbf{a}_1' \\ \mathbf{a}_1'^\top \mathbf{R}_{21} \mathbf{a}_1 & \mathbf{a}_1'^\top \mathbf{R}_{22} \mathbf{a}_1' \end{pmatrix}. \tag{6.2}$$

The diagonal elements of $\mathbf{C}_{y_1, y_1'}$ give the variances of y_1 and y_1' while each of the off-diagonal elements gives the covariance between the two linear combinations. In the extreme case, when the genes in the two groups are uncorrelated, $\mathbf{R}_{12} = \mathbf{0}_{k_1 \times (p-k_1)}$,

and $C_{y,y'}$ is block-diagonal.

Associated with the square symmetric matrix in (6.2), we need a single-number summary that involves all the elements of the matrix. Such a number can be given by the determinant of the matrix (sometimes called the generalized variance):

$$\begin{aligned} f_2(\mathbf{s}_1, \mathbf{s}_2) &= |C_{y,y'}| \\ &= (\mathbf{a}_1^\top \mathbf{R}_{11} \mathbf{a}_1) \times \left(\mathbf{a}_1'^\top \mathbf{R}_{22} \mathbf{a}_1'\right) - \left(\mathbf{a}_1^\top \mathbf{R}_{12} \mathbf{a}_1'\right)^2 . \end{aligned} \tag{6.3}$$

This number is then used for choosing the 'best' partition. That is, we choose a partition of $\mathbf{s}$, say $\mathbf{s}_1^*$ and $\mathbf{s}_2^*$, among all possible partitions for which the value of $f_2(\mathbf{s}_1^*, \mathbf{s}_2^*)$ in (6.3) is the largest. As p is large, the eigenvalue-eigenvector pairs can be efficiently obtained from the SVD of the partitioned data matrices $\mathbf{X}_1 \in \mathbb{R}^{n \times k_1}$ and $\mathbf{X}_2 \in \mathbb{R}^{n \times (p-k_1)}$.

At the 'best' solution of the first stage of the algorithm, $\mathbf{s}_1^*$ gives the first cluster of k_1 genes. To form the next cluster, the ordering and partitioning procedures are repeated on the remaining $p - k_1$ vector of genes contained in $\mathbf{s}_2^*$. In general, the data matrix of $p - \sum_{i=0}^{k} k_i$ ordered genes is used at the ith stage of the partitioning procedure with $k_0 = 0$.

The whole procedure of gene-ordering and partitioning continue until no further clustering is possible or until one gets the required number of clusters (whichever comes first). But, if one is interested in a required number of clusters which exceeds the one obtained at the termination of the procedure, then it is possible to repeat the whole algorithm on one or more of the resulting clusters. At the extreme case, one can continue the procedure until each gene makes a cluster.

6.1.3 Initializing a cluster

The two initializing genes are important factors in forming a new cluster. As a result, we need to set some rule. One possible rule is that a pair of genes initializes a new cluster if the absolute value of their correlation coefficient is not less than a certain (nonnegative) threshold value, say r_0. Thus, a minimum absolute correlation coefficient (MACC) could be chosen based on the problem under investigation. This step helps to avoid unnecessary grouping together of uncorrelated genes.

The cluster-initializing issue is also related to the criteria for terminating the clustering algorithm. If the correlation coefficient between a pair of initializing genes is less than r_0 at a particular stage, then the gene ordering and partitioning procedures terminates. Suppose that clusters of sizes k_i $(i = 1, 2, \ldots, m)$ are already formed by the first m clustering stages, and that the absolute correlation coefficient between the two initializing genes for the next stage is less than r_0. Then, the $q = (p - \sum_{i=1}^{m} k_i)$ unclustered genes either make one cluster each, leading to the total number of clusters being $(m+q)$, or may be regarded as 'noise' (or outliers) so that they might not make clusters.

As a gene expression data set often contains a huge amount of noise, one of the challenges in gene clustering is related to the extraction of useful information from background noise. The possibility that the semi-partition method could filter out noise may be one merit of the method over the k-means approach, which forces each of the genes to join a cluster. However, such genes may not necessarily be noise and, instead, require further investigation.

The following algorithm summarizes the semi-partition clustering procedure:

1. Let $\mathbf{s}_0 = \{1, 2, ..., p\}$ contain the initial indices of the genes. Then, find the pair of indices from $\mathbf{s}_0$, such that the $n \times 2$ matrix of the corresponding genes has the largest absolute correlation coefficient among all pairs from $\mathbf{s}_0$. If the absolute correlation coefficient is less than a pre-specified value r_0, stop the algorithm and return the result; otherwise, denote this pair by $\mathbf{s}_1$ and let $\mathbf{s}_2 \leftarrow \mathbf{s} \backslash \mathbf{s}_1$ ($\mathbf{s}$ without $\mathbf{s}_1$). Find $f_2(\mathbf{s}_1, \mathbf{s}_2)$ according to (6.3) above.

2. Identify a gene from $\mathbf{s}_2$ which, together with the two genes from $\mathbf{s}_1$, form a $n \times 3$ matrix with the largest sum of correlation coefficients among all other genes from $\mathbf{s}_2$. Update $\mathbf{s}_1$ and $\mathbf{s}_2$ by removing this gene from $\mathbf{s}_2$ and inserting into $\mathbf{s}_1$. Find new $f_2(\mathbf{s}_1, \mathbf{s}_2)$, compare with the previous, and keep the largest.

3. Continue removing a gene from $\mathbf{s}_2$ and inserting it into $\mathbf{s}_1$, based on the maximal value of (6.1), until $\mathbf{s}_2$ gets only one gene.

4. Identify the partition, say $\mathbf{s}_1^*$ and $\mathbf{s}_2^*$, that gives the largest value $f_2(\mathbf{s}_1^*, \mathbf{s}_2^*)$ of the criterion (6.3). Then $\mathbf{s}_1^*$ gives the first cluster of genes.

5. To get the next cluster, repeat the ordering and partitioning on the vector of genes $\mathbf{s}_2^*$ (i.e. $\mathbf{s}_0 \leftarrow \mathbf{s}_2^*$ and go to step 1, but now p denoting the length of $\mathbf{s}_2^*$).

6. Continue the algorithm until a required number of clusters is obtained or until no further clustering is possible.

At the end of the algorithm, the variables that are not in any cluster will be considered as single-cluster variables.

6.1.4 Evaluating the clustering algorithm

The Rand index (Rand, 1971; Yeung and Ruzzo, 2001) is a known method for evaluating clustering algorithms. It helps to measure the similarity of two clusterings of the same data. For a given p-vector $\mathbf{x}$ of genes, consider a pair of clusterings of genes $\mathcal{C}_1 = \{C_{11}, C_{12}, \ldots, C_{1k_1}\}$ and $\mathcal{C}_2 = \{C_{21}, C_{22}, \ldots, C_{2k_2}\}$. This could be the case where the two clusterings are obtained from applying two different methods to the same data set, and hence C_{ij} denotes the jth cluster obtained from the ith method. Rand (1971) proposed a measure of similarity between $\mathcal{C}_1$ and $\mathcal{C}_2$, denoted by $S(\mathcal{C}_1, \mathcal{C}_2)$, as

$$S(\mathcal{C}_1, \mathcal{C}_2) = 1 - \frac{0.5 \times \left[\sum_{i=1}^{k_1}\left(\sum_{j=1}^{k_2} p_{ij}\right)^2 + \sum_{j=1}^{k_2}\left(\sum_{i=1}^{k_1} p_{ij}\right)^2\right] - \sum_{i=1}^{k_1}\sum_{j=1}^{k_2} p_{ij}^2}{\binom{p}{2}} \tag{6.4}$$

where p_{ij} denotes the number of genes simultaneously in the ith cluster of $\mathcal{C}_1$ and in the jth cluster of $\mathcal{C}_2$. This can simply be defined as the proportion of concordant gene pairs in two partitions among all possible gene pairs (Thalamuthu *et al.*, 2006)). If a denotes the number of pairs of elements that are in the same set in $\mathcal{C}_1$ and in the same set in $\mathcal{C}_2$, and b denotes the number of pairs of elements that are in different sets in $\mathcal{C}_1$ and in different sets in $\mathcal{C}_2$, then the Rand index is simply given by

$$S(\mathcal{C}_1, \mathcal{C}_2) = \frac{a+b}{\binom{p}{2}}.$$

The overlappings between $\mathcal{C}_1$ and $\mathcal{C}_2$ can be summarized in a contingency table where p_{ij} denotes the number of common genes of groups C_{1i} and C_{2j}: $p_{ij} = |C_{1i} \bigcap C_{2j}|$. The values of the Rand index lie between 0 (when the two data clusters do not agree on

any pair of genes) and 1 (when the data clusters are exactly the same).

Thus, the performance of a clustering method can be evaluated based on the similarity of the resulting clusters with the 'true clusters' of a data set. But, this requires a data set whose number of true clusters is known. In Section 6.3, we use the Rand index for assessing the performances of the semi-partition and the k-means methods based on a data set having five (known) clusters.

6.2 Cluster-based sparse principal components

The motivation to the construction of the cluster-based sparse PCs is already given in Section 5.2 of the previous chapter. This section briefly describes some points related to the construction of cluster-based sparse PCs based on the semi-partition clustering method.

6.2.1 Constructing cluster-based sparse principal components

Assume that the variables are already grouped into k clusters and that the jth cluster is composed of q_j variables, for $j = 1, \ldots, k$ and $\sum_{j=1}^{k} q_j = p$. Let $\boldsymbol{\omega}_j$ be the $q_j \times 1$ vector containing the indices of the original variables clustered into the jth cluster in ascending order, i.e. $\omega_{1,j} \leq \omega_{2,j} \leq \ldots \leq \omega_{q_j,j}$. Define the following $p \times q_j$ indicator matrices $\mathbf{G}_j$, for $j = 1, \ldots, k$: $\mathbf{G}_j$ has 1 at its position $(\omega_{l,j}, l)$ for $l = 1, \ldots, q_j$ and 0 otherwise. Then $\mathbf{X}_j = \mathbf{X}\mathbf{G}_j$ is the $n \times q_j$ data submatrix corresponding to the jth cluster and let $\mathbf{X}_j = \mathbf{U}_j\mathbf{L}_j\mathbf{A}_j^\top$ be its singular value decomposition (SVD). Denote by $\mathbf{v}_1^{(j)}$ the singular vector of $\mathbf{A}_j$ corresponding to the largest singular value in $\mathbf{L}_j$. Then, the $p \times k$ matrix $\mathbf{V}$ of cluster-based sparse principal component (CSPC) loadings is

formed as follows:

$$\mathbf{V} = [\mathbf{G}_1\mathbf{v}_1^{(1)}|\mathbf{G}_2\mathbf{v}_1^{(2)}|...|\mathbf{G}_k\mathbf{v}_1^{(k)}] \, .$$

6.2.2 Goodness-of-fit

The goodness-of-fit for the cluster-based method can be measured using the cumulative proportion of variances explained by the CSPCs, compared to the cumulative proportion of the variances of the data matrix. If the matrix of loadings $\mathbf{V}$ is obtained based on the matrix $\mathbf{R}$, then the diagonal elements of the $k \times k$ symmetric matrix

$$\mathbf{S}_k = \mathbf{V}^\top \mathbf{R} \mathbf{V}$$

give the variances explained by the k CSPCs. But, the sparse components are correlated to each other, and hence the adjusted variances (Zou *et al.*, 2006) are used as a better measure of goodness-of-fit. If $\mathbf{F}_k$ denotes the upper triangular matrix of the Cholesky factorization of $\mathbf{S}_k$, then the adjusted variances are given by the squared diagonal elements of $\mathbf{F}_k$.

6.2.3 Number of cluster-based sparse principal components

The number of CSPCs depends on the number of clusters. But, as indicated in the previous chapter, there is no hard and fast rule for choosing the number of clusters in cluster analysis. There are few attempts to determine the number of clusters in microarray gene expression data (McLachlan *et al.*, 2004, Section 4.12).

One simple constraint affecting the number of clusters (and hence the number of CSPCs) might be the determination of the threshold r_0 while initializing a cluster. The value of r_0 may vary depending on the type of data under consideration. For

gene expression data, for instance, the genes are often highly correlated to each other and a relatively higher value should be set to r_0. Depending on the value of r_0, some genes may be left unclustered. This may not be the case for small data sets with $n > p$, where each of the unclustered variables may form a cluster, leading to sparse components each with one nonzero loading variable. But, r_0 is introduced simply to avoid the clustering together of uncorrelated variables and may not be a necessity for the semi-partition algorithm to run. The semi-partition algorithm in Section 6.1 can continue until no further clustering is possible, without requiring to set r_0. In this case, the algorithm finds k clusters, which number is unknown *a priori* and is a result of a particular optimal ordering/partitioning process. This implies that the number of sparse components should not always be prescribed in advance, say based on the scree plot of the original data.

It is also possible to constrain the number of clusters to a required number k' where $k' \leq k$. This number k' is supposed to govern the dimension and the sparsity of the components. However, if one is interested in k' clusters where $k < k' \leq p$, then it is possible to repeat the semi-partition algorithm on one or more of the clusters themselves. For this purpose, the next possible cluster to be partitioned into two further clusters could be the one which gives the "maximum weakest-link" between the partitions. That is, if $f_{2(i)}$ denotes the maximal value of (6.3) corresponding to the partitioned correlation matrix of the ith cluster ($i = 1, 2, \ldots, k$), then the cluster that should be divided into further clusters is the one with the largest value of $f_{2(i)}$.

6.2.4 Semi-partition versus k-means

If the k-means method is used for clustering variables, then correlations between the variables can serve as the measure of (dis)similarity in which $(1-correlation)$ is proportional to the squared distance. In many situations, the k-means method outperforms many other existing clustering methods. But, one drawback of the k-means method is that the solution depends highly on the initial values. In addition, it requires to fix the required number k of clusters *a priori*. It is included in this chapter for comparison with the semi-partition, with the number of semi-partition clusters functioning as the value of k.

Recall that the ultimate goal of the semi-partition clustering method is to form clusters in such a way that the cumulative adjusted variances explained by the corresponding CSPCs is maximized. The clustering method proposed in Section 6.1 is designed to produce clusters of noticeable size differences, and the CSPCs corresponding to the first few clusters explain large proportion of variances. On the other hand, the k-means clusters "tend" to have relatively similar sizes, which leads the corresponding first few CSPCs to have smaller cumulative variances than those based on the semi-partition method. In general, the CSPCs based on the two clustering methods can be compared using the percentage of cumulative adjusted variances explained by the respective CSPCs (see Section 6.2.2).

6.3 Application

In this section, CSPCs based on the semi-partition clustering approach is illustrated using different kinds of data sets, and the results are compared with that of the k-

means method. The semi-partition clustering algorithm is based on a MATLAB code written by the author, while the k-means is based on the MATLAB function `kmeans` with correlation as the distance parameter. In addition, the CSPCs corresponding to the semi-partition clusters are compared with the sparse PCs based on Witten *et al.* (2009).

6.3.1 Simple data sets ($n > p$)

Synthetic data

Here, we consider again the artificial data set generated in Section 4.3, simply to illustrate the behaviour of the clustering approach to sparse principal components. Recall that the data set contains 10 variables x_i ($i = 1, 2, \ldots, 10$), which fall into 4 groups (or clusters) by construction: $\{x_1\}$, $\{x_2, x_3\}$, $\{x_4, x_5, x_6\}$, and $\{x_7, x_8, x_9, x_{10}\}$.

Suppose that n observations are generated, and let $\mathbf{x}_i$ denotes the n-vector of observations for the ith variable. Put $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{10}]$, an $n \times p$ matrix of observations. Then, application of the semi-partition clustering method to $\mathbf{X}$ with $n = 100$ results in the four clusters of variables, $\{x_7, x_8, x_9, x_{10}\}$, $\{x_4, x_5, x_6\}$, $\{x_2, x_3\}$, and $\{x_1\}$. This conforms with the groups of variables given by the initial construction. For a required number of four clusters, the hierarchical and the k-means clustering methods also give the same set of clusters as the semi-partition method (not shown here).

Actually, the number of variables involved in this simulated data set is so small that the clusters can easily be identified from the dendrogram of the hierarchical clusterings. However, it is not easy to identify the required clusters if the number of variables is too large, as with gene expression data sets (to be illustrated later).

One desirable property of the semi-partition method worth noting is that it forms sets of clusters in a sequential way, which may help to construct only the first few clusters of variables while leaving some variables unclustered. Such a procedure may have an extra advantage of keeping away those variables with outlying observations from joining the first few clusters. For instance, we may be interested in only the first cluster for the synthetic data, which is found to be $\{x_7, x_8, x_9, x_{10}\}$. At this point, the other variables are considered unclustered. If the interest is in the first two clusters, then we need to search for the second cluster using only the unclustered variables, without affecting the first one. This results in $\{x_4, x_5, x_6\}$ as the second cluster, still leaving the variables x_1, x_2, and x_3 unclustered. The same procedure continues if more clusters are required until all variables have been clustered.

If the semi-partition method is considered as a useful clustering method, the above issue is especially useful in contrast to the k-means clustering method, which forms the k clusters simultaneously and hence requires re-initializing the centroids for each change in the required number of clusters. The value of k is usually unknown before hand, but the k-means method forces each variable to join one of the k clusters. For instance, if $k = 2$ is used in the synthetic data, the k-means clusters become $\{x_7, x_8, x_9, x_{10}\}$ and $\{x_1, x_2, x_3, x_4, x_5, x_6\}$. Such a result might be affected by outlying values.

The CSPCs corresponding to the four clusters of variables is the same as the IPCs given in Chapter 5. Table 5.4 gives the loadings and the cumulative percentage of adjusted variances for the first four standard PCs and the corresponding CSPCs. The CSPCs are much sparser (and simpler to interpret) than the PCs while accounting for almost as much cumulative variances as the PCs. The k-means (with $k = 4$), the

hierarchical and the semi-partition clustering methods each gives the same CSPCs.

Alate Adelges data

Jeffers (1967) used a data set from Alate Adelges (winged aphids), in which 19 variables are measured from 40 individual alate adelges, with the purpose of determining the number of distinct taxa that were present at a particular habitat. He uses principal component analysis to get guidance on the number of taxa, and he tried to interpret the first four PCs. The 19 variables are body length (x_1), body width (x_2), fore-wing length (x_3), hind-wing length (x_4), number of spirales (x_5), number of antennal segment I (x_6), number of antennal segment II (x_7), number of antennal segment III (x_8), number of antennal segment IV (x_9), number of antennal segment V (x_{10}), number of antennal spines (x_{11}), leg length – tarsus (x_{12}), leg length – tibia (x_{13}), leg length – femur (x_{14}), rostrum (x_{15}), ovipositor (x_{16}), number of ovipositor spines (x_{17}), anal fold (x_{18}) and number of hind-wing hooks (x_{19}). Their correlation matrix, given in Jeffers (1967), is used here.

For any MACC values in $0 \leq r_0 \leq 0.8$, the semi-partition clustering method groups the 19 variables into four clusters as $\{x_1, x_2, x_3, x_4, x_6, x_7, x_8, x_9, x_{10}, x_{12}, x_{13}, x_{14}, x_{15}, x_{19}\}$, $\{x_5, x_{16}, x_{17}\}$, $\{x_{11}\}$ and $\{x_{18}\}$. The first cluster contains the majority of the variables due to the high degree of correlation among them.

The semi-partition clusters can be compared with those of the hierarchical and the k-means methods. Figure 6.1 gives the dendrogram of the hierarchical clusterings based on the single, complete, and average (also called group average) linkages. For a required number of four clusters, both the complete and the average linkages give the same result as that of the semi-partition, but the single linkage gives slightly different

results: three clusters each with a single variable (from $\{x_{11}\}$, $\{x_{17}\}$, and $\{x_{18}\}$) and the fourth cluster containing all the remaining variables. Indeed, it is emphasized (Everitt and Dunn, 2001, Section 6.2.3) that the single linkage is usually the least satisfactory method compared to the complete and average linkages. On the other hand, the k-means method, with a required number of four clusters, results in $\{x_1, x_2, x_3, x_4, x_6, x_7, x_9, x_{10}, x_{12}, x_{13}, x_{14}, x_{15}\}$, $\{x_5, x_{16}, x_{17}\}$, $\{x_{11}, x_{18}\}$ and $\{x_8, x_{19}\}$. These are quite different from that of the semi-partition.

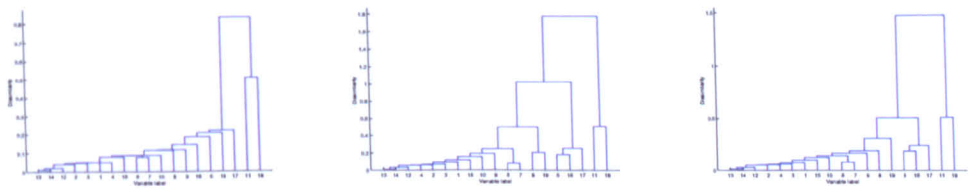

Figure 6.1: *Dendrograms for the Alate Adelges data: Single-linkage (Left), complete-linkage (middle) and average-linkage (Right).*

The first CSPC based on the semi-partition method is almost a general index of the size of the individuals, while the other three CSPCs are interpreted with respect to the nonzero-loading variables in the respective sparse components. The four CSPCs explain 78.52% of the cumulative adjusted variances. The hierarchical clustering method based on each of the complete and average linkages give the same result as that of the semi-partition, but the one based on the single linkage explains 78.32%. On the other hand, the four CSPCs based on the k-means method explain 70.26% of the total variation, a higher loss in information than with those based on the semi-partition clustering. Table 6.1 gives the component loadings and the cumulative adjusted variances explained by the components based on each of the ordinary PCA and the cluster-based methods.

Table 6.1: *Loadings and percentage of cumulative adjusted variances (CAV) for the first four PCs and CSPCs based on semi-partition (SP) and k-means (KM), Alate Adelges data. Empty cells have zero loadings.*

	PCs				CSPCs (SP)				CSPCs (KM)			
Variable	1	2	3	4	1	2	3	4	1	2	3	4
x_1	.25	-.03	.02	.07	.27				.29			
x_2	.26	-.07	.01	.10	.28				.29			
x_3	.26	-.03	-.05	.07	.28				.29			
x_4	.26	-.09	.03	.00	.28				.29			
x_5	.16	.41	-.19	-.62		.58				.58		
x_6	.24	.18	.04	-.01	.25				.27			
x_7	.25	.16	.00	.02	.27				.29			
x_8	.23	-.24	.05	.11	.25							.71
x_9	.24	-.04	.17	.01	.26				.27			
x_{10}	.25	.03	.10	-.02	.27				.28			
x_{11}	-.13	.20	.93	-.17			1.00				.71	
x_{12}	.26	-.01	.03	.18	.28				.30			
x_{13}	.26	-.03	.08	.20	.28				.30			
x_{14}	.26	-.07	.12	.19	.28				.30			
x_{15}	.25	.01	.07	.04	.27				.29			
x_{16}	.20	.40	-.02	.06		.59				.59		
x_{17}	.11	.55	-.15	.04		.57				.57		
x_{18}	-.19	.35	.04	.49				1.00			.71	
x_{19}	.20	-.28	.05	-.45	.22							.71
CAV(%)	73.0	85.4	89.4	92.0	63.9	72.8	76.8	78.5	56.9	65.3	68.5	70.3

6.3.2 Gene expression data ($p \gg n$)

A gene expression dataset collects together the expression values from a series of DNA microarray experiments, with each column representing an experiment. There are several thousand rows representing individual genes, and tens of columns representing

samples. Two different gene expression data sets are considered here. The first is used especially for assessing the performance of the semi-partition clustering method, while the second is used for constructing clusters and computing the corresponding CSPCs. Comparison of the CSPCs with other sparse components is also based on the second data. Due to the large number of genes, the CSPCs are not presented here.

The Yeast data

The yeast cell cycle data (henceforth referred to as the yeast data), presented in Yeung and Ruzzo (2001), contains the fluctuation of the expression levels of 384 genes over 17 time points. The data, publicly available at `http://faculty.washington.edu/kayee/pca/`, is first log-transformed and then each gene vector is mean-centered and normalized to have length 1.

As indicated by Yeung and Ruzzo (2001), the yeast data is a subset of a larger data set initially employed by Cho *et al.* (1998), who originally categorized the genes into five phases of cell cycles. The phase of each gene is given together with the raw data in the link given above. The number of genes corresponding to each of the five phases are found to be 67, 135, 75, 52, and 55. These can serve as the true clusters to which the clusterings from each of the semi-partition and the k-means methods are compared based on the Rand index (Section 6.1.4). Denote the five true clusters by serial numbers 1 to 5.

Both the semi-partition and the k-means methods are applied to the yeast data, with a required number of five clusters. Let C_1, C_2, C_3, C_4 and C_5 denote the five clusters. The contingency table in Table 6.2 gives the number of genes in these clusters and the corresponding true clusters (given as rows). Each cell in the table consists of

two values, given outside and inside bracket, corresponding to the number of genes for the semi-partition and the k-means clusters, respectively. Each element in the body of the table corresponds to the p_{ij} defined in (6.4). The last row of the table gives the cluster-sizes for the semi-partition (and the k-means) clusters, while the last column gives the cluster-sizes for the true clusters.

The Rand index, computed from Table 6.2, corresponding to the semi-partition is found to be 0.79, while the Rand index corresponding to the k-means is 0.80. The k-means method performed slightly better than the semi-partition, though the values are not far from each other. On the other hand, the cumulative percentage of adjusted variances explained by the five semi-partition CSPCs (45.8%) is slightly higher than the one explained by the five k-means CSPCs (44.4%).

Table 6.2: *Contingency table for the number of genes in the semi-partition, k-means (in brackets) and true clusters, Yeast data*

	Semi-partition (k-means) clusters					
True Cl	C_1	C_2	C_3	C_4	C_5	Total
1	36(48)	5(6)	0(0)	13(0)	13(13)	67
2	13(17)	120(114)	2(4)	1(0)	1(0)	135
3	1(2)	32(29)	26(35)	7(9)	7(0)	75
4	0(0)	0(0)	21(23)	7(26)	24(3)	52
5	1(1)	0(0)	1(0)	6(19)	47(35)	55
Total	51(68)	157(149)	50(62)	34(54)	92(51)	384

The Alon data

The Alon data (Alon *et al.*, 1999) corresponds to the gene expression measurements publicly available from `http://microarray.princeton.edu/oncology/`. The data

matrix consists of 2000 genes with minimal intensity across 62 samples, 40 tumor and 22 normal colon tissue samples. But, some of the genes in the data set are duplicated (i.e., there are more than one different expression sequences). As such genes are highly correlated, only one of the sequences, which has the largest standard deviation, is considered. This procedure reduces the number of unique genes to 1909. Thus, our final Alon data matrix consists of a mean-centered and normalized vectors of the (natural) logarithm of $p = 1909$ genes in $n = 62$ samples.

Application of the semi-partition algorithm to the Alon data with a value of r_0 in $[0.5, 0.85]$ results in 10 clusters. The cluster-sizes range from 647 genes for the first cluster to 8 genes for the last cluster. The clusters include the last one formed from the set of remaining genes when the algorithm terminates due to the fact that the maximum absolute correlation coefficient (MACC) between a pair of genes is less than 0.5 (set arbitrarily). It might be possible that each of the genes in this last cluster convey a unique information, and thus needs further investigation, rather than putting them as a 'cluster'. However, for the sake of comparison with other similar clustering methods, such as k-means, we consider the ten clusters as the final result. The cluster-sizes for each of the semi-partition and the k-means clusters are shown in Figure 6.5.

Heat maps are used to look for similarities between genes and between samples. They are most effective if rows and columns are ordered so as to allow these patterns to be identified. To see if the semi-partition clusters of genes are genuine with respect to the expression patterns, two heat maps are plotted, ranging from green (negative) to red (positive): one before clustering, and another after clustering the genes. The two heat maps (depicted in Figure 6.2) are plotted using a heat map builder freely available

at `http://ashleylab.stanford.edu/tools_scripts.html`. The rows of the heat map represent the genes and the columns represent the samples (with columns 1–40 for the tumor and columns 41–62 for the normal samples). The genes in the first heat map are given in the alphabetical order of their codes, which has no connection with the expression levels. In the second heat map, the genes in each cluster are given in the order they joined the corresponding cluster, and hence co-expressing genes come closer to each other within a cluster. The cluster boundaries (viewed horizontally) are visible from this heatmap compared to that of the unclustered genes. The result could be a simple and visual demonstration that the method has performed well in clustering the genes.

Most of the regions in the heat map of the unclustered genes look green compared to the clustered one, due to the loss in resolution resulting from the large number of genes. This feature hides the fact that the two heat maps just represent a rearrangement of rows. Figure 6.3 gives a cut-down version of the two heat maps, involving 100 genes from each of the first five clusters. This helps to see the distribution of the colors in both plots, except the effect of clustering.

Figure 6.4 gives the dendrogram of the genes based on the average-linkage hierarchical clustering method. The horizontal axis of the plot represents the different genes while the vertical axis gives the measure of dissimilarity between pairs of genes, which are obtained by the `pdist` function in MATLAB with 'correlation' as a distance parameter. It might not be simple to identify a required number of clusters from such a dense dendrogram. The usual trend in approximating a required number of clusters from the mergers of a dendrogram may not give the real situation. For the dendrogram in Figure 6.4, a required number of 10 clusters corresponds to the number of

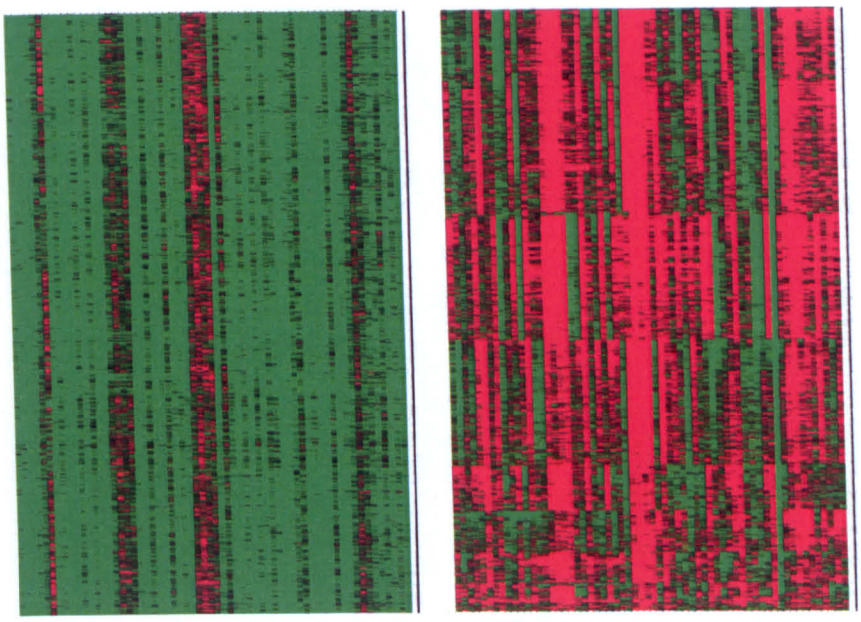

Figure 6.2: *Heat map before clustering (Left) and after clustering (Right) using all genes, Alon data*

mergers at the dissimilarity measure of about 0.65. But, this leads to the situation where almost all genes fall in one cluster.

However, a general picture of possible clusterings can be visualized from the dendrogram, especially from the left part of the plot. We designated the first few of such 'blocks' of genes by three letters A, B, and C. It is found that most of the genes in category A correspond to the second semi-partition cluster, while those under category B correspond to the first semi-partition cluster. In general, most of the genes designated on the dendrogram by the three letters are found in the first three semi-partition clusters. Thus, the semi-partition method may help to find the plausible clusters by keeping away some noisy observations from the first few clusters.

The CSPCs from the semi-partition and the k-means methods are computed from the data matrix of each cluster and then sorted in descending order of their adjusted

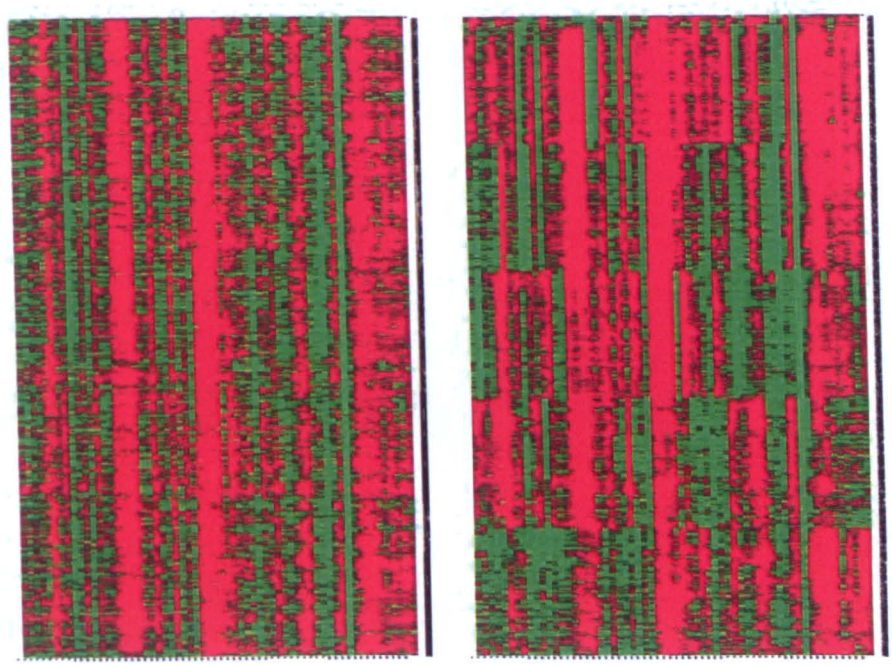

Figure 6.3: *Heat map before clustering (Left) and after clustering (Right) using 100 genes from each of five clusters, Alon data*

variances. Then, the two methods are compared with respect to the number of nonzero-loadings (the left-hand plot in Figure 6.5) and the cumulative proportion of adjusted variances explained (the right-hand plot in Figure 6.5). The left-hand plot shows in general that the k-means method tends to produce clusters of similar sizes compared to the semi-partition method. But, the k-means is doing better than semi-partition with respect to sparseness for the first few components. From the right-hand plot, the semi-partition method leads to CSPCs explaining higher cumulative proportion of adjusted variances than those of the k-means method, given the same number of clusters in both methods.

For the Alon data set, the semi-partition algorithm takes 5.5 minutes to give both the clusters and the corresponding CSPCs on an Intel(R) Pentium 4 computer with 3.2GHz CPU and .99 GB of Ram. Compared to the size of the data, the speed seems

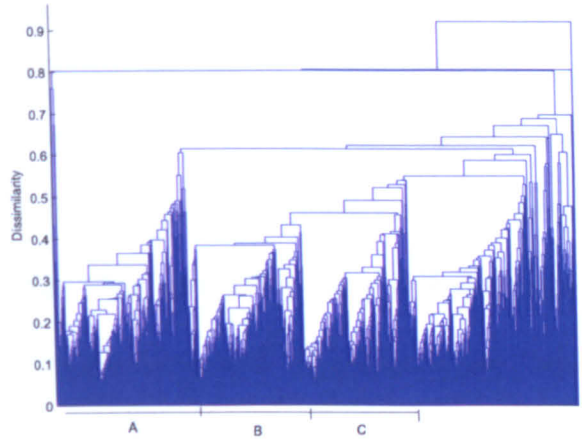

Figure 6.4: *Average-linkage dendrogram for the genes, Alon data*

fair.

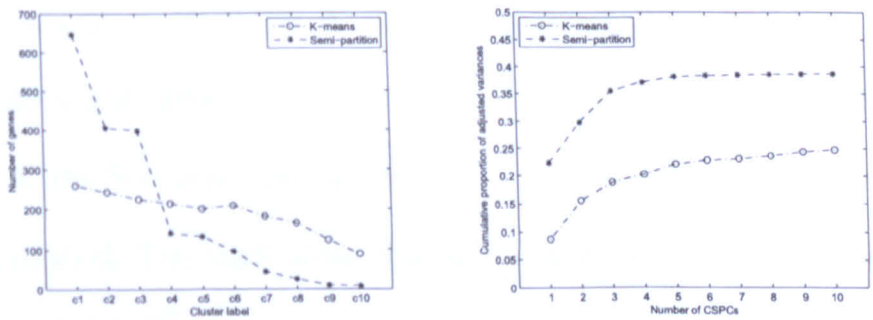

Figure 6.5: *The semi-partition versus k-means clustering methods with respect to the number of genes per cluster (Left) and cumulative proportion of adjusted variances explained by the corresponding CSPCs (Right), Alon data*

6.3.3 Semi-partition versus gene-shaving

Gene-shaving (Hastie *et al.*, 2000) aims to identify few small-sized clusters of genes each with similar expression patterns. On the other hand, the semi-partition clustering algorithm is designed to identify clusters of genes with similar expression levels, but

each cluster having a rather large size. As a result, a gene-shaving cluster could be a subset of the semi-partition cluster. Gene-shaving may not be considered as a clustering method, but our basis for comparison with the semi-partition is attributed to the fact that both methods are looking for sets of co-expressing genes. In addition, both methods have connections with PCA – the gene-shaving algorithm involves PCA in that a cluster of genes are those with high correlation with a PC, while the semi-partition method works towards approximating a standard PC with a cluster-based sparse PC.

In this section, we make a simple comparison between the two methods using the Alon data. McLachlan *et al.* (2004) used this data set to demonstrate the (unsupervised) gene-shaving method. For the sake of comparison, we consider the genes in the four gene-shaving clusters given on p.181 of McLachlan *et al.* (2004). Table 6.3 relates the genes in the four gene-shaving clusters to those obtained by the semi-partition clustering method. The result shows that all the genes in the first gene-shaving cluster fall into the third semi-partition cluster. Similarly, all the genes in the second gene shaving cluster fall into the second semi-partition cluster. On the other hand, almost all the genes in both the third and the fourth gene-shaving clusters (except one gene, which belongs to another semi-partition cluster) are grouped into the first semi-partition cluster.

To see if further partitioning into two of the first semi-partition cluster can identify the third and the fourth gene-shaving clusters, we repeated the gene ordering and partitioning procedures on this cluster. The result showed that all the genes in the fourth gene-shaving cluster (among those already in the first semi-partition cluster in the first stage) fall into one of the new semi-partition clusters while all but five genes in

the third gene-shaving cluster fall into another cluster. Thus, the semi-partition helps to identify a wider range of co-expressing genes whenever this is deemed important.

There are, however, noticeable differences between the semi-partition and the gene-shaving methods. Due to the ultimate objective of constructing sparse principal components, the semi-partition method requires to group all or the majority of the genes into non-overlapping clusters, and hence each cluster may contain a relatively large number of genes. In addition, the number of clusters may not necessarily be fixed a priori, and the cluster-sizes are automatically decided by the algorithm itself. In contrast, the gene-shaving method is designed to extract only a small number of co-expressing clusters of genes. It requires fixing the number of clusters a priori and the optimal cluster size is estimated using the 'gap statistic' (Tibshirani *et al.*, 2001). In addition, the genes in the gene-shaving clusters may be overlapping.

Table 6.3: *Cluster membership in the semi-partition (SP) of the genes clustered by gene-shaving (GS), Alon data*

GS #1	SP #	GS #2	SP #	GS #3	SP #	GS #4	SP #
L02426	3	R34876	2	U21914	1	U27143	1
M26697	3	T57686	2	R15814	1	R49231	1
T51023	3	T60437	2	D26018	1	R43913	1
R43914	3	T57468	2	R33367	1	X72727	1
M84326	3	X12466	2	D14689	1	R22779	1
M88279	3	M29065	2	L10413	1	L19437	1
M22382	3	T52642	2	U14588	1	T69748	1
M14200	3	H24030	2	R53936	1	T70595	4
T69446	3	T56244	2	D26067	1	T92259	1
T93589	3	H05899	2	D13641	1	H88250	1
T84049	3	T63591	2	R09468	1	X68194	1
T40674	3	H69869	2	R71585	1	H09719	1
R60859	3	T65758	2	D21260	1	D14043	1
H89087	3	U02493	2	D13627	1	D17400	1
R37428	3	M21339	2	U18062	1	H38185	1
R16156	3			L10911	1	H42127	1
D00761	3			R27813	1	X87838	1
				M90104	1	L19437	1
				X01060	1	D15057	1
				R50864	1	U20998	1
				X16135	1		

6.3.4 Cluster-based versus other sparse methods

Witten *et al.* (2009) argue that their sparse principal component (SPC) method is superior to the sparse principal component analysis by Zou *et al.* (2006) with respect to some basic properties. In this section, the cluster-based sparse principal component

(CSPC) method is compared with the SPC method using the Alon data set. The comparison involves the level of sparsity and the cumulative proportion of variances explained by the components. First, some adjustments are made as follows, for a fair comparison between the two methods.

As discussed in Chapter 4, the SPC function in R (Witten *et al.*, 2009) uses as one of its required arguments the sum of the absolute values (**sumabsv**) of the loadings in a sparse component. This value is assumed to measure the level of sparsity and is set by the user. On the other hand, the CSPC method is designed in such a way that the components involve non-overlapping genes. This feature is not shared by the SPC method. However, **sumabsv** can be calculated from components of the CSPC method and the SPC components can be made as non-overlapping as possible, so that both the SPC and CSPC methods are put on a similar footing for a fair comparison. This can be accomplished using the following procedures [this is similar to that of Section 4.3, except the first step]:

a. Run the semi-partition clustering algorithm and compute $c_1, \ldots, c_m$ from the corresponding sparse components where c_i is the sum of absolute values of the p elements in the ith CSPC with the ith largest variance.

b. Run the SPC algorithm in R with **sumabsv** $= c_1$ in order to get component 1. Then subtract out this first component and perform SPC on the residuals to get component 2, with **sumabsv** $= c_2$. That is, if $\mathbf{v}_1$ denotes the first SPC, computed based on the data matrix $\mathbf{X}_1$, then the second sparse component $\mathbf{v}_2$ is computed on the residual data matrix $\mathbf{X}_2 \equiv \mathbf{X}_1 - \mathbf{X}_1\mathbf{v}_1\mathbf{v}_1^\top$.

c. Repeat this procedure until a required number $k'(\leq k)$ of the first SPCs have

been obtained. In general, the ith sparse component $\mathbf{v}_i$ is computed based on the residual data matrix $\mathbf{X}_i \equiv \mathbf{X}_{i-1} - \mathbf{X}_{i-1}\mathbf{v}_{i-1}\mathbf{v}_{i-1}^{\mathrm{T}}$ for $i = 1, \ldots, k'$, with $\mathbf{X}_0 = \mathbf{X}$ and $\mathbf{v}_0$ being a vector of zeros.

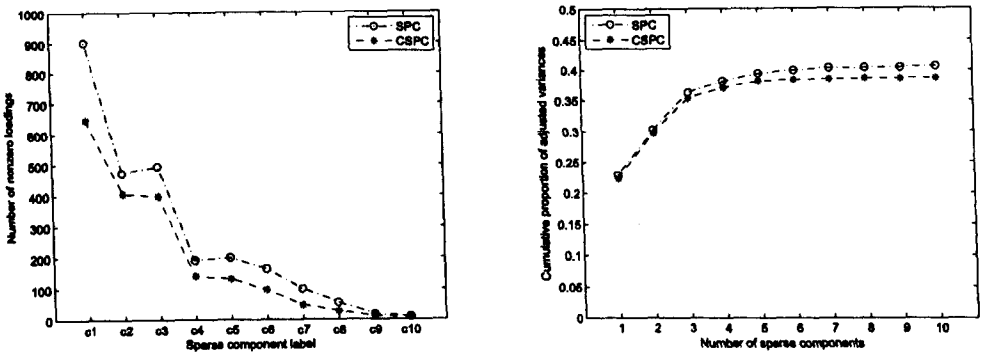

Figure 6.6: *Comparison of components from CSPC and SPC methods with respect to sparsity (left-hand plot) and cumulative proportion of adjusted variances explained (right-hand plot) when the respective components are allowed to have the same* **sumabsv** *values, Alon data.*

The two methods give varying results on the level of sparsity (measured by the number of zero-loading genes) and the cumulative proportion of adjusted variances as shown in Figure 6.6. The SPC method depends highly on the value of **sumabsv**: higher values result in less sparse components (but with higher cumulative variance explained), and vice versa. The level of sparsity for both methods is generally decreasing with an increase in the explained adjusted variances. The components from the CSPC method are in general sparser than the SPC method, but explain smaller cumulative percentage of adjusted variances. Thus, the choice between the two methods depends on the preference between sparsity of the components and the cumulative variances explained by the components.

6.4 Summary

Like Chapter 5, this chapter involves clustering approach as a pre-processing step to simplified components. Interpretable sparse PCs are constructed from the data matrix of clusters of variables. The simplicity in the interpretation of the cluster-based sparse PCs is attributed to the level of sparsity and the non-overlappingness of the components with respect to the nonzero-loading variables, which are gained through clustering.

A two-stage clustering approach, called semi-partition, is proposed for this purpose. It is designed especially for data sets with a larger number of variables than observations, such as gene expression data sets. This is in comparison with the weighted-variance clustering method proposed in Chapter 5, which is limited only to those data sets with smaller number of variables than observations. The semi-partition clustering algorithm is designed in such a way that the percentage of cumulative adjusted variance explained by few of the resulting cluster-based sparse PCs is maximized.

Comparison of the cluster-based sparse PCs using artificial and real data sets show that sparse components from semi-partition clustering approach explain higher percentage of cumulative adjusted variance than those based on existing clustering methods, such as the k-means method. Furthermore, the level of sparsity differs among the two types of cluster-based sparse components. Each sparse component based on the k-means method tends to have similar number of nonzero-loading variables, while those based on the semi-partition method involve varying number of nonzero loading variables. The latter may be preferred when one needs to consider only few sparse PCs explaining higher percentage of cumulative adjusted variances, as in the ordinary

PCA.

Chapter 7

Penalized varimax

7.1 Introduction

So far, especially in the last three chapters, we were more interested in new techniques for simplifying the interpretation of principal components. This chapter, however, is slightly different in that it is concerned with simplifying the interpretation of factors in factor analysis. In particular, it deals with penalizing the simple structure varimax rotation criterion so that the resulting rotated factors are easily interpreted.

Analytic rotation methods have a long history in exploratory factor analysis. Browne (2001) gives a very complete and comprehensive overview of the field. Details can be found in the papers cited there and in the standard texts on factor analysis. See, for example, Harman (1976) and Mulaik (1972).

A common weakness of all analytical methods for simple structure rotation is that the rotated factors are usually unequally loaded, which may spoil their interpretation. For instance, the quartimax rotation tends to produce solutions with a dominating factor (Harman, 1976; Knüsel, 2008). Such a factor is dominated by larger loadings,

and hence has a much higher sum of squared loadings compared to the remaining factors. On the other hand, the varimax solution has a tendency towards equal sum of squared loadings for all factors. This probably explains the great success of the varimax criterion. Unfortunately, the existing varimax algorithms do not try to achieve this optimal property explicitly. Recently Knüsel (2008) has shown that, indeed, in theory the varimax solution should have equal sum of squared loadings for all factors. As is well-known (Harman, 1976; Mulaik, 1972), ideally the varimax criterion is maximized when there is a single unit loading per factor and all the rest are 0s, which also implies equal (to 1) sum of squares per factor. Thurstone's simple structure criteria (Thurstone, 1947, p.335) also suggest equidistributed zeros across the rows and the columns of the rotated loading matrix.

In this chapter, a modified varimax criterion is introduced by attaching a penalty term to the original varimax objective function. The penalty term explicitly controls the size of the column sums of squared loadings, by 'equi-distributing the load' from the overloaded factors to the less-loaded factors. As a result, the penalized varimax solution has equal sum of squared loadings for all factors. The penalized varimax approach is designed as a supplement to the classical varimax procedure which treats problems with unsatisfactory simple structure possibly caused by uneven sum-of-squares per factor.

The chapter is organized as follows. A formulation of the varimax rotation problem, and a list of the most popular algorithms for its solution, are given in Section 7.2. Definitions of the penalized varimax problem are proposed in Section 7.3. It is solved by a matrix algorithm making use of the projected gradient method (Jennrich, 2001; Trendafilov, 2006). The matrix algorithm directly finds an orthogonal rotation matrix

to produce the penalized varimax solution (Section 7.3.2). If the penalty term is switched off, the algorithm simply turns into a standard varimax rotation.

The method is applied to three benchmark data sets: the five socio-economic variables (Harman, 1976, p.135), the 24 psychological tests data (Harman, 1976, p.123 and p.215) and Thurstone's box data (Thurstone, 1947, p.370). The results are compared to the classical varimax solutions. It is demonstrated that if the application of the penalized varimax is reasonable, it can provide clearer simple structure than the standard varimax solution.

7.2 Varimax criterion

Varimax (Kaiser, 1958) is the most popular method for analytical rotation in factor analysis. Let $\mathbf{A}$ be the initial $p \times k$ matrix of factor loadings and $\mathbf{B} := \mathbf{AQ}$ be an orthogonally rotated factor loadings matrix. The variance of the squared loadings of the jth rotated factor is:

$$V_j = \sum_{i=1}^{p} b_{ij}^4 - \frac{1}{p}\left(\sum_{i=1}^{p} b_{ij}^2\right)^2 . \tag{7.1}$$

For a given sum of squared loadings, the variance V_j will be large when there are few large squared loadings and all the rest are near zero. The variance V_j will be small when all squared loadings have nearly same value. The varimax rotation problem (Kaiser, 1958) is to find a $k \times k$ orthogonal matrix $\mathbf{Q}$ such that the total variance of all k factors is maximized, i.e. maximize

$$V = \sum_{j=1}^{k} V_j = \sum_{j=1}^{k}\left[\sum_{i=1}^{p} b_{ij}^4 - \frac{1}{p}\left(\sum_{i=1}^{p} b_{ij}^2\right)^2\right] . \tag{7.2}$$

The original algorithm to find the varimax rotation $\mathbf{Q}$ proposed by Kaiser (1958)

makes use of successive planar rotations of all possible $k(k-1)/2$ pairs of factors, such that each pair has maximum variance.

The varimax rotation problem can be defined in matrix form as follows (Magnus and Neudecker, 1988; Sherin, 1966). Compose the matrix:

$$\mathbf{S}(\mathbf{Q}) = \mathbf{C}^\top \mathbf{M}_p \mathbf{C} \text{ with } \mathbf{M}_p = \left(\mathbf{I}_p - \frac{\mathbf{1}_p \mathbf{1}_p^\top}{p} \right) , \tag{7.3}$$

where $\mathbf{C} = \mathbf{B} \odot \mathbf{B}$ and $\odot$ denotes the Hadamard (elementwise) matrix product. If $\mathbf{S}$ in (7.3) is divided by $p-1$, it presents the (sample) covariance matrix of the squared orthogonally transformed factor loadings $\mathbf{B}$. Then, the varimax problem is to maximize the following objective function (criterion):

$$V(\mathbf{Q}) = \text{trace } \mathbf{S}(\mathbf{Q}) , \tag{7.4}$$

over all possible orthogonal rotations $\mathbf{Q} \in \mathcal{O}(k)$, i.e.,

$$\max_{\mathbf{Q} \in \mathcal{O}(k)} \text{trace } \mathbf{S}(\mathbf{Q}) . \tag{7.5}$$

A number of different algorithms are available for solving the varimax problem. See, for example, Jennrich (1970), Kaiser (1958), Magnus and Neudecker (1988), Mulaik (1972), Sherin (1966), and ten Berge (1984).

7.3 Varimax with equal column sums of squares

For an orthogonal rotation $\mathbf{Q}$, the sum of the squared initial loadings equals the sum of the squared rotated loadings. That is,

$$\text{trace}(\mathbf{A}\mathbf{A}^\top) = \text{trace}(\mathbf{A}\mathbf{Q}\mathbf{Q}^\top\mathbf{A}^\top) = \text{trace}(\mathbf{B}\mathbf{B}^\top). \tag{7.6}$$

The aim of this chapter is to construct an algorithm that finds loadings maximizing the varimax criterion subject to having equal sums of squares across the factors. In other words, the rotated loadings should possess the property:

$$\mathbf{b}_1^\top \mathbf{b}_1 = \ldots = \mathbf{b}_k^\top \mathbf{b}_k \ . \tag{7.7}$$

To achieve this one should solve the original varimax problem (7.5) subject to the additional constraint (7.7). An alternative way to impose the additional constraint (7.7) on the varimax solution is to modify the varimax criterion by adding a term penalizing the deviation from (7.7). This modified varimax problem is called the penalized varimax. The next section deals with the construction of such a penalty term.

7.3.1 Penalizing unequal column sums of squares of B

Consider Lagrange's identity:

$$k\sum_{j=1}^{k} x_j^2 = \left(\sum_{j=1}^{k} x_j\right)^2 + \sum_{1\le j<q\le k} (x_j - x_q)^2 \ , \tag{7.8}$$

where $x_1, x_2, \ldots x_k$ are any real numbers. Clearly, if $\sum_{j=1}^{k} x_j$ is constant, then $\sum_{j=1}^{k} x_j^2$ is minimized when $x_j = x_q$ for any $1 \le j < q \le k$, and conversely. Substituting $x_j = \mathbf{b}_j^\top \mathbf{b}_j$ in (7.8) and using (7.6), we see that

$$\sum_{j=1}^{k} (\mathbf{b}_j^\top \mathbf{b}_j)^2 = \mathbf{1}_p^\top \mathbf{C}\mathbf{C}^\top \mathbf{1}_p \ge \frac{(\text{trace}(\mathbf{A}^\top \mathbf{A}))^2}{k} \ , \tag{7.9}$$

equality holding if and only if $\mathbf{b}_1^\top \mathbf{b}_1 = \ldots = \mathbf{b}_k^\top \mathbf{b}_k$. The inequality is the result of the fact that the second term on the right-hand side of (7.8) is non-negative. The inequality can also be inferred from the algebra on the next page.

Define the penalty term as

$$\mathcal{P}(\mathbf{Q}) = \mathbf{1}_p^\top \mathbf{C}\mathbf{C}^\top \mathbf{1}_p - \frac{(\text{trace}(\mathbf{A}^\top \mathbf{A}))^2}{k} , \tag{7.10}$$

which is a nonnegative continuous function of the rotation matrix $\mathbf{Q} \in \mathcal{O}(k)$. As $\mathcal{O}(k)$ is compact, there exists $\mathbf{Q}$ at which $\mathcal{P}(\mathbf{Q})$ achieves its minimum value of 0. Thus, $\mathcal{P}(\mathbf{Q})$ penalizes unequal column sums of squares of the rotated loading matrix $\mathbf{B}$ and vanishes when and only when $\mathbf{b}_1^\top \mathbf{b}_1 = ... = \mathbf{b}_k^\top \mathbf{b}_k$. One can easily see that, in fact, $\mathcal{P}(\mathbf{Q})$ penalizes the total deviation of all column sums of squares of $\mathbf{B}$ from their mean value. That is, for $j = 1, ..., k$, put $b_{\cdot j}^2 = \sum_i^p b_{ij}^2$ and $b_{\cdot\cdot}^2 = \frac{1}{k}\sum_j^k b_{\cdot j}^2 = \frac{1}{k}\sum_j^k (\sum_i^p b_{ij}^2)$. Then,

$$\begin{aligned}
\sum_{j=1}^{k} (b_{\cdot j}^2 - b_{\cdot\cdot}^2)^2 &= \sum_{j=1}^{k} \left(\sum_i^p b_{ij}^2 - \frac{\sum_i^k \sum_j^p b_{ij}^2}{k} \right)^2 \\
&= \sum_{j=1}^{k} \left(\mathbf{b}_j^\top \mathbf{b}_j - \frac{\text{trace}\mathbf{B}^\top \mathbf{B}}{k} \right)^2 \\
&= \sum_{j=1}^{k} (\mathbf{b}_j^\top \mathbf{b}_j)^2 - 2\frac{\text{trace}\mathbf{B}^\top \mathbf{B}}{k} \sum_j^k \mathbf{b}_j^\top \mathbf{b}_j + k \left[\frac{\text{trace}\mathbf{B}^\top \mathbf{B}}{k} \right]^2 \\
&= \mathbf{1}_p^\top \mathbf{C}\mathbf{C}^\top \mathbf{1}_p - \frac{1}{k} (\text{trace}\mathbf{A}^\top \mathbf{A})^2
\end{aligned}$$

since $\sum_j^k \mathbf{b}_j^\top \mathbf{b}_j = \text{trace}\mathbf{B}^\top \mathbf{B} = \text{trace}\mathbf{A}^\top \mathbf{A}$.

7.3.2 Penalized varimax criterion

Consider the following penalized varimax criterion:

$$PV(\mathbf{Q}) = \text{trace } \mathbf{C}^\top \mathbf{M}_p \mathbf{C} - \mu \mathcal{P}(\mathbf{Q}) , \tag{7.11}$$

where μ is a large positive number and the penalty term $\mathcal{P}(\mathbf{Q})$ is given in (7.10). As with the standard varimax, by maximizing the PV criterion (7.11), the loadings are forced to get either small values around 0 or values near 1 or -1, but having as equal as

possible sums of squared loadings of all factors. The importance of the penalty term is controlled by varying μ. Low values of μ will result in solutions close to the original variamax ones, while large values of μ can suppress entirely the varimax maximization and result in $\mathbf{B}$ with equal column sums of squares.

As the penalty term $\mathcal{P}(\mathbf{Q})$ in (7.11) contains a constant term which will not be affected by the maximization process, it seems more reasonable and cheaper to work with the following penalized varimax criterion:

$$PV(\mathbf{Q}) = \text{trace } \mathbf{C}^\top \mathbf{M}_p \mathbf{C} - \mu \mathbf{1}_p^\top \mathbf{C}\mathbf{C}^\top \mathbf{1}_p \, . \tag{7.12}$$

The penalized varimax criterion $PV(\mathbf{Q})$ in (7.12) is in matrix form. The penalized varimax problem requires solving the following constrained maximization problem:

$$\max_{\mathbf{Q} \in \mathcal{O}(k)} PV(\mathbf{Q}) \, . \tag{7.13}$$

Problem (7.13) can be readily solved by the orthogonal rotation algorithm `varimaxP` based on the dynamical system approach proposed by Trendafilov (2006). For this purpose, the gradient of $PV(\mathbf{Q})$ is needed, which, in turn, requires a smooth approximation of the penalty term. The same results are obtained by iterative implementation of the gradient projection algorithm of Jennrich (2001). As the penalized varimax function $PV(\mathbf{Q})$ is more complicated, alternatively one can rely on the derivative-free version of the gradient projection algorithm (Jennrich, 2004).

Straightforward manipulations (Magnus and Neudecker, 1988) give the gradient of trace $\mathbf{C}^\top \mathbf{M}_p \mathbf{C}$ as:

$$4\mathbf{A}^\top \left(\mathbf{B} \odot (\mathbf{M}_p \mathbf{C}) \right) \, , \tag{7.14}$$

and the gradient of the penalty in (7.12), as:

$$4\mathbf{A}^\top \left(\mathbf{B} \odot (\mathbf{1}_p \mathbf{1}_p^\top \mathbf{C}) \right) \, . \tag{7.15}$$

Then, the gradient of the objective function $PV(\mathbf{Q})$ is:

$$4\mathbf{A}^\top \left\{ \mathbf{B} \odot \left[\left(\mathbf{M}_p - \mu \mathbf{1}_p \mathbf{1}_p^\top \right) \mathbf{C} \right] \right\} . \tag{7.16}$$

This gradient will be used, along with the objective function $PV(\mathbf{Q})$, in solving (7.13) using the dynamical system approach (Trendafilov, 2006)

7.4 Numerical examples and comparisons

Simple artificial data

An idea about the behaviour of the penalized varimax approach can be given by the following small artificial example. Consider the loading matrix given in the first two columns of Table 7.1. Strictly speaking, such a loading matrix has nearly perfect simple structure, and does not need rotation at all. The only unsatisfied condition for perfect simplicity is that the first column has less 0s than factors (Thurstone, 1947, p.335). Applying the standard varimax algorithm has no effect, the loadings are left unrotated. Then the `varimaxP` algorithm is applied for different μ. For $\mu \in [0, 2.5]$, `varimaxP` also leaves the loadings unrotated. After further increasing μ, the penalty term becomes more important than the varimax term, as seen in the next columns of Table 7.1. Finally, one ends up with the worst possible simple structure solution given in the last two columns of Table 7.1. This example is artificial and unlikely to happen in practice, but it shows that the penalized varimax approach should not be applied automatically. Using inappropriate μ may lead to unsatisfactory loadings. In reality, loading matrices composed by 0s and $\pm$1s only are very hard to find and impossible to achieve by orthogonal rotation. In general, the penalized varimax approach is expected

to produce factors with balanced contributions to the total variance of the solution, while retaining well its simple structure.

Table 7.1: *Limitations of the penalized varimax.*

	Initial loadings		varimax (MATLAB)		varimaxP ($\mu = 2.6659$)		varimaxP ($\mu = 2.6669$)		varimaxP ($\mu = 2.7$)	
Var	I	II	I	II	I	II	I	II	I	II
1	1	0	1	0	1.00	.06	.86	-.50	.71	-.71
2	1	0	1	0	1.00	.06	.86	-.50	.71	-.71
3	0	1	0	1	.06	1.00	.50	.86	-.71	-.71
s.s.	2	1	2	1	2	1	1.75	1.25	1.5	1.5
varimax	1.33		1.33		1.31		.33		.00	

Data from five socio-economic variables

The simple structure rotation of the first two principal components of the five socio-economic variables (Harman, 1976, p.135) gives a more realistic example illustrating the same potential problem with the penalized varimax approach. The standard varimax solution is given in the first two columns of Table 7.2. The loadings have pretty good simple structure. The column sums of squares are of quite similar magnitude ($2.15/2.52 = .85$). After these observations are made, the application of the penalized varimax approach seems unreasonable: there is not much room to either improve or spoil the varimax solution. Nevertheless, for illustration purposes, the `varimaxP` algorithm is applied with three different $\mu = 1, 5, 10$ and they are depicted in Table 7.2. Increasing μ gives more nearly equal column sum of squares while losing little of the original simplicity of the $\mu = 0$ solution.

Table 7.2: *Factor loadings for the five socio-economic variables from two varimax algorithms.*

	varimax (MATLAB)		varimaxP ($\mu = 1$)		varimaxP ($\mu = 5$)		varimaxP ($\mu = 10$)	
Var	I	II	I	II	I	II	I	II
1	.01	.99	-.03	.99	-.10	.99	-.12	.99
2	.94	-.00	.94	.04	.94	.10	.93	.12
3	.13	.98	.09	.99	.02	.99	-.00	.99
4	.82	.45	.80	.49	.77	.54	.75	.56
5	.97	-.00	.97	.04	.96	.11	.96	.13
s.s.	2.52	2.15	2.47	2.20	2.40	2.27	2.37	2.30
varimax	1.8684		1.8560		1.7885		1.7496	

24 psychological tests data

Three varimax rotations (without Kaiser's normalization) are applied to the maximum likelihood solution for the 24 psychological tests (Harman, 1976, p.215), called for short 24HH data. The value of the varimax criterion for this initial solution is 0.6249. The first four columns of Table 7.3 are obtained by the classical varimax rotation algorithm based on plane rotations and implemented in MATLAB (MATLAB, 2009). For 100 random starts, the algorithm results in the same optimal loadings with no local maxima. The value of the varimax criterion is 2.5110. For the same number of random starts, the varimaxP algorithm without penalty ($\mu = 0$) produces exactly the same loadings (not depicted) as the MATLAB one and no local maxima.

The first factor of the varimax solution has relatively big sum of squares loadings. Then the varimaxP algorithm with $\mu = 20$ is applied and the solution given in the

last four columns in Table 7.3. The value of the penalized varimax criterion for this solution is -654.9085, and the value of the varimax criterion is 2.2326. For the HH24 data, $\frac{1}{4}(\text{trace}\mathbf{A}^\top\mathbf{A})^2 = 32.8570$, this lower bound in (7.9) being achieved by the penalty term for the depicted solution, as $\mathbf{1}_k^\top\mathbf{C}\mathbf{C}^\top\mathbf{1}_k = 32.8570$. In other words, the sum of squared loadings of the factors are equal here. For 100 random starts, the `varimaxP` algorithm with $\mu = 20$ produces no local maxima.

Table 7.3: *Factor loadings for HH24 data from two varimax algorithms.*

	varimax (MATLAB)				varimaxP ($\mu = 20$)			
Var	I	II	III	IV	I	II	III	IV
1	.25	.15	**.68**	.13	.08	**.68**	.17	.25
2	.17	.06	**.43**	.08	.07	**.43**	.07	.16
3	.21	-.05	**.55**	.10	.08	**.56**	-.04	.19
4	.30	.07	**.50**	.05	.18	**.53**	.09	.16
5	**.76**	.21	.12	.07	**.69**	.23	.27	.23
6	**.80**	.07	.12	.16	**.72**	.23	.13	.32
7	**.83**	.15	.12	-.01	**.76**	.25	.21	.17
8	**.61**	.23	.29	.06	**.51**	.37	.28	.21
9	**.84**	.05	.11	.15	**.76**	.23	.11	.32
10	.17	**.85**	-.08	.08	.09	-.07	**.85**	.13
11	.22	**.53**	.13	.31	.09	.11	**.54**	.38
12	.05	**.70**	.26	.03	-.07	.24	**.70**	.08
13	.24	**.50**	**.45**	.02	.10	**.47**	**.52**	.13
14	.25	.12	.03	**.53**	.14	-.00	.12	**.57**
15	.18	.11	.11	**.50**	.06	.06	.10	**.53**
16	.16	.08	.40	**.51**	-.01	.35	.07	**.57**
17	.20	.26	.06	**.54**	.07	.01	.25	**.58**
18	.10	.35	.31	**.42**	-.07	.25	.34	**.47**
19	.20	.17	.23	.34	.08	.21	.18	.40
20	**.44**	.12	.36	.26	.31	.39	.14	.37
21	.23	**.43**	.39	.17	.09	.39	**.44**	.26
22	**.43**	.12	.36	.26	.30	.39	.14	.37
23	**.44**	.23	**.47**	.18	.29	**.50**	.26	.32
24	**.41**	**.51**	.15	.23	.29	.17	**.53**	.33
ss	4.35	2.69	2.62	1.81	2.87	2.86	2.86	2.86
varimax	2.5110				2.2326			

Loadings greater than .4 in the solutions depicted in Table 7.3 are shown in bold typeface. The simple structure of the `varimaxP` ($\mu = 20$) solution is clearer than the one following from the standard varimax (`MATLAB`). In fact, the `varimaxP` simple struc-

ture almost exactly matches (except $b_{21,3}$) the simple structure obtained in (Browne, 2001, p.133) from *oblique* rotation.

Thurstone's 26 box data

The same varimax rotation algorithms (without Kaiser's normalization) are applied to Thurstone's 26 Box problem (Thurstone, 1947), called for short 26 Box data. The initial solution to be analyzed below comprises the first three principal components extracted by Cureton and Mulaik (1975) from the correlation matrix of the 26 Box data (Thurstone, 1947, p.370). The value of the varimax criterion for this initial principal component solution is 6.1017.

The first three columns in Table 7.4 are the varimax solution for the 26 Box problem obtained by the `MATLAB` algorithm (MATLAB, 2009). The maximum of the varimax objective function is 6.2365. No local maxima are found within 100 random runs. The `varimaxP` algorithm without penalty ($\mu = 0$) produces exactly the same loadings (not depicted) as the `MATLAB` ones and no local maxima for 100 runs.

The first factor of the standard varimax solution of the 26 Box data (first three columns of Table 7.4) has considerably large sum of squares loadings. This is a clear indication to apply the penalized varimax approach. The next three columns in Table 7.4 are obtained by the `varimaxP` algorithm with $\mu = 20$. The value of the penalized varimax criterion for this solution is -4298.6981, and the value of the varimax criterion is 5.5309. For the 26 Box data, $\frac{1}{4}(\text{trace}\mathbf{A}^\top\mathbf{A})^2 = 215.2114$, which is achieved by the penalty term for the depicted solution, i.e. $\mathbf{1}_k^\top\mathbf{C}\mathbf{C}^\top\mathbf{1}_k = 215.2115$. In other words, the sum of squared loadings of the factors are equal here. For 100 runs, the `varimaxP`

algorithm with $\mu = 20$ produces no local maxima.

The standard varimax solution does not reveal any clear simple structure for the 26 Box data: the loadings look to be a complete mess. It is clear that the penalized varimax solution has much more structured loadings. Moreover, it provides a kind of 'negative' (as in photography) simple structure in the 26 Box data.

The difficulties experienced with revealing simple structure in the 26 Box data is a notorious problem with the varimax criterion. This is not surprising because the weighted varimax solution of Cureton and Mulaik (1975), which reveals the simple structure in the 26 Box data, has a varimax value of only 5.3746. The orthogonal minimum entropy solution of the 26 Box problem reported by Browne (2001) also reveals its simple structure and has varimax value 5.4370. Clearly, all these 'successful' solutions are local maxima for the varimax criterion.

Such local maxima are also obtained by the penalized varimax approach. While experimenting with `varimaxP`, it was observed that new local maxima emerge when using a very short integration step. One can get rid of them by increasing the required convergence accuracy, say from 10^{-4} to 10^{-7}, between two consecutive varimax values. For 100 random runs, only one local maximum of the penalized varimax criterion was observed with value of -4298.86, which for the varimax criterion gives 5.37. Ironically, just this solution reconstructs well the simple structure of the 26 Box data and is given in the last three columns of Table 7.4.

Table 7.4: *Factor loadings for 26 Box data from two varimax algorithms.*

	varimax(MATLAB)			varimaxP($\mu = 20$)			varimaxP(local)		
Vars	I	II	III	I	II	III	I	II	III
x_1	.61	-.22	.74	-.25	.71	.64	.98	-.04	.14
x_2	.69	.68	-.04	.63	-.09	.73	.28	.92	.12
x_3	.83	-.33	-.42	.73	.66	-.04	.14	.23	.94
x_1x_2	.81	.35	.44	.26	.34	.89	.77	.60	.13
x_1x_3	.90	-.38	.17	.32	.88	.33	.68	.10	.72
x_2x_3	.91	.22	-.34	.86	.32	.38	.20	.71	.66
$x_1^2x_2$	.78	.11	.59	.06	.54	.83	.91	.35	.16
$x_1x_2^2$	.80	.52	.22	.46	.16	.85	.57	.78	.14
$x_1^2x_3$	.83	-.35	.41	.10	.86	.46	.83	.03	.53
$x_1x_3^2$	.00	-.41	-.04	.52	.91	.23	.56	.17	.91
$x_2^2x_3$	.86	.41	-.26	.82	.17	.51	.22	.83	.49
$x_2x_3^2$	.91	.03	-.39	.85	.45	.24	.18	.57	.79
x_1/x_2	-.06	-.79	.61	-.70	.69	-.13	.55	-.83	.07
x_2/x_1	.06	.79	-.61	.70	-.69	.13	-.55	.83	-.07
x_1/x_3	-.15	.15	.96	-.77	.01	.61	.70	-.17	-.68
x_3/x_1	.15	-.15	-.96	.77	-.01	-.61	-.70	.17	.68
x_2/x_3	-.10	.95	.28	-.02	-.71	.70	.08	.68	-.73
x_3/x_2	.10	-.95	-.28	.02	.71	-.70	-.08	-.68	.73
$2x_1 + 2x_2$	.80	.43	.37	.32	.26	.89	.70	.67	.11
$2x_1 + 2x_3$	.90	-.40	.12	.34	.89	.28	.64	.09	.76
$2x_2 + 2x_3$	.91	.22	-.32	.85	.33	.40	.22	.72	.65
$(x_1^2 + x_2^2)^{1/2}$	.79	.42	.36	.32	.26	.87	.69	.66	.12
$(x_1^2 + x_3^2)^{1/2}$	.88	-.38	.10	.36	.86	.27	.61	.10	.74
$(x_2^2 + x_3^2)^{1/2}$	.90	.23	-.29	.82	.32	.41	.24	.71	.63
$x_1x_2x_3$	.98	.08	.11	.54	.57	.61	.62	.54	.55
$(x_1^2 + x_2^2 + x_3^2)^{1/2}$	.96	.14	.01	.61	.49	.57	.52	.59	.56
s.s.	14.79	5.55	5.08	8.47	8.47	8.47	8.47	8.47	8.47
varimax	6.2365			5.5309			5.3700		

Chapter 8

Discussion and future research directions

This thesis aims to contribute towards simplification of the interpretation of new variables (or components), especially in PCA, while reducing a p-dimensional multivariate data to a lower dimension, say k ($\ll p$). Thus, the main objective is to propose simple and fast techniques of constructing interpretable components. In addition, the proposed techniques aim to contribute to the determination of the number k of PCs (and hence the number of interpretable components) to retain.

The sparse biplots (sBarse) component analysis in Chapter 4 proposes a very efficient method to simplify the interpretation of PCs, with several advantages over existing ones. It may avoid subjective judgement in ignoring small-magnitude loadings in the 'classical' way for simple interpretation of PCs. The loadings of each sBarse component take values from $\{0, \pm c\}$ with $(0 < c \leq 1)$. The method is designed in such a way that the variables associated with each sBarse component do not overlap, leading to clearer interpretation of the components. The first k ($\leq p$) sBarse compo-

nents have in total p nonzero loadings and are used for interpretation, the remaining $p - k$ components being identically zero. Thus, the method may help to suggest the number k of components that account for the majority of the variation in the original variables. This can be used as an alternative to existing methods, such as the scree plot and the cumulative percentage of variance explained by components, which may involve subjective judgment.

The examples given in Chapter 4 illustrate that the sBarse method gives very good solutions compared to existing, usually more complicated, methods to obtain simplified (sparse) components (SCs). For the Pitprop data, for instance, it gives the sparsest orthogonal components among the available SCs by other methods. In addition, it can be readily applied to data sets with $p \gg n$. For the gene expression data set, the sBarse method results in sparser orthogonal components than that of the sparse method by Witten *et al.* (2009). The components also explain a good proportion of cumulative variance. Each sBarse component contains a group of nonzero-loading genes, which do not overlap with those in the other sBarse components. This leads to a clearer and easier interpretation of the components, compared to those computed by other sparse methods.

The sBarse method may not be able to produce sparse loadings for each $k = 1, \ldots, p$. However, this is not a serious problem as we are usually interested in the minimal number of SCs accounting for as much variation as possible. Another difficulty might be the lack of clear guidance for the α's to consider in order not to skip the best sparse solution. However, taking only a few of them suffices, as for certain discretization, different intervals of α's correspond to identical sBarse solutions. An alternative option is to narrow the search interval around the current solution on each

consecutive stage of the algorithm. This option may help to speed up the algorithm.

Chapter 5 proposes a cluster-based approach to interpretable principal components (IPCs), in which sparse components are constructed from clusters of variables. The motivation for this approach is that important variables comprising a certain component are more correlated with each other than with other variables. As a result, variables are grouped into clusters using a certain objective criterion, and an IPC is constructed from each cluster. The construction is in such a way that only the variables in the corresponding cluster take nonzero loadings, while the remaining variables are assigned zero loadings and the IPC is sparse. The nonzero loadings are obtained from the eigenvector of the correlation matrix of the variables in the corresponding cluster.

Existing clustering methods might not be suitable for constructing the IPCs due to their design and purpose. In addition, these methods often fix the number k of clusters a priori. For this reason, a new weighted-variance clustering method is proposed which results in k clusters, a number which could either be given as required or automatically obtained from the algorithm. The latter option results in the 'best' sets of clusters among all possible clusters. This may also help in approximating the number k of IPCs, which can be inferred from the cluster plot. Application to synthetic and real data sets demonstrates that the IPCs based on the weighted-variance clustering method explain a higher percentage of cumulative adjusted variances, a desirable property in the ordinary PCs, compared to those based on existing clustering methods.

An additional benefit of the proposed clustering method is that it can be used as an alternative pre-processing step in variable selection. That means, once the p variables are grouped into k clusters, a representative variable can be retained from

each cluster. This is in contrast to the other variable selection methods, such as the 'principal variables' by McCabe (1984) and other methods discussed by Jolliffe (1972).

The cluster-based sparse method in Chapter 5 is designed only for data sets with a larger number of observations than variables. This restriction led us to propose another clustering method, called the semi-partition method, especially designed for data sets with a larger number of variables than observations. Chapter 6 proposes a cluster-based sparse PCs method based on the semi-partition method. The method is developed with microarray gene expression data sets as the main target, although it is applicable in general to any type of data set, including those with $n > p$.

The procedures in the semi-partition clustering method look in some way like that of an ordinary (partitioning) clustering method, but it is based entirely on a different criterion due to the objective. Existing clustering methods are solely concerned with grouping the variables or observations based on some measure of similarities/dissimilarities, whereas the semi-partition clustering method is targeted at forming cluster-based sparse principal components (CSPCs) which share some properties with the ordinary PCs, such as maximizing variances. Despite these differences, the CSPCs based on both the semi-partition and existing methods are computed and compared with respect to their adjusted explained variances. The result can show the need for proposing a new more appropriate clustering method.

Comparison of the CSPCs based on the semi-partition method with that of the k-means method, using real gene expression data, shows that the CSPCs based on the former method explain a higher proportion of cumulative adjusted variances than those based on the latter one. In addition, the CSPCs based on the semi-partition method show varying levels of sparsity (as measured by the cluster-sizes) while those

based on the k-means method show a relatively homogeneous level. This difference is crucial when one is interested in considering only the first few CSPCs with high cumulative percentage of adjusted variances, in which case those based on the semi-partition method are preferred. The CSPCs are also compared with the sparse principal components by Witten *et al.* (2009), and are found to perform well with respect to the level of sparsity at comparable levels of adjusted variance explained by the components.

One limitation of the cluster-based sparse method is that it depends on the quality of the semi-partition clusters, which in turn depends on the factors affecting the clustering method, such as the initialization of a new cluster. Choosing a threshold value of the correlation coefficient for a pair of cluster-initializing variables (or genes) might pose subjectivity, especially for clusters forming at the later stages of the algorithm. This might be a drawback only if interest is in grouping each and every variable into a specific cluster (like the k-means method), and if the method is used to determine the number of possible clusters. However, this is not the usual case with the semi-partition method, especially for large data sets such as those arising in gene expression studies, as only the first few clusters (and hence the first few CSPCs) are required to approximate the data matrix in a reduced dimension.

Here, it is possible to compare the methods in Chapters 4 to 6 with respect to some features. A common characteristics of the simple component methods given in each of the chapters is that the resulting components are non-overlapping with respect to their nonzero-loading variables. In addition, no constraint is involved to make the components sparse (unlike, say, methods using the LASSO constraint, such as SCoTLASS). However, tuning parameters are used in some cases. For instance, the sBarse solution can be affected by the value of α used in the biplot factor. Similarly,

the semi-partition clustering approach to sparse component requires setting a minimum absolute correlation coefficient in initializing a new cluster. Both Chapter 5 and Chapter 6 involve clustering approach to sparse components, but the weighted-variance clustering method of Chapter 5 tends to be more objective in determining the number of components (using cluster plot) than that of the semi-partition clustering method of Chapter 6. The nonzero-loadings of each sBarse component are equal (and hence adds simplicity to its interpretation) while each simple component resulting from the clustering methods involves unequal nonzero-loadings.

Chapter 7 is somewhat different to the other preceding chapters in that it deals with simplifying the interpretation of factors in factor analysis rather than PCs. A penalized version of the well-known varimax orthogonal rotation method is proposed which produces loadings having equal sums of squares for all factors. Such factors are balanced and may give more adequate interpretation for some data. The penalized varimax is proposed as a supplement (companion) procedure to the standard varimax, especially for rotating factor solutions with considerably overloaded factor. This is illustrated by the 26 Box data in Table 7.4.

We end by briefly indicating some possible future research directions. The cluster-based PCs method proposed in Chapter 5 is limited to the case where the number of variables is less than the number of observations. However, the technique might possibly be extended to any type of data. This possibility is not explored in this thesis, and shall be done in the future. Another direction with respect to cluster-based method could be the use of overlapping clusters, which may result in sparse PCs with overlapping variables. It might be necessary to obtain sparse components in which a particular variable can assume nonzero loadings in more than one component.

This and other issues can be addressed with in possible extension of the sBarse method (Chapter 4), such as introducing a tuning parameter(s) measuring the importance of a variable and considering more than one component for which a particular variable is relatively important.

Bibliography

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., and Levine, A. J. (1999). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, **96**, 6745–6750.

Anaya-Izquierdo, K., Critchley, F., and Vines, K. (2010). Orthogonal simple component analysis. *to appear in The Annals of Applied Statistics*.

Bernaards, C. A. and Jennrich, R. I. (2005). Gradient projection algorithms and software for arbitrary rotation criteria in factor analysis. *Educational and Psychological Measurement*, **65**, 676–696.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford.

Borg, I. and Groenen, P. (1997). *Modern Multidimensional Scaling: Theory and Applications*. Springer-Verlag, New York.

Browne, M. W. (2001). An overview of analytic rotation in exploratory factor analysis. *Multivariate Behavioral Research*, **36**, 111–150.

Cadima, J. and Jolliffe, I. T. (1995). Loadings and correlations in the interpretations of principal components. *Journal of Applied Statistics*, **22**, 203–214.

Cadima, J. and Jolliffe, I. T. (2001). Variable selection and the interpretation of principal subspaces. *Journal of Agricultural, Biological, and Environmental Statistics*, **6**, 62–79.

Cattell, R. B. (1966). The scree test for the number of factors. *Mult. Behav. Res.*, **1**, 245–276.

Chin, K., Devries, S., Fridlyand, J., Spellman, P., Roydasgupta, R., Kuo, W., Lapuk, A., Neve, R., Qian, Z., Ryder, T., Chen, F., Feiler, H., Tokuyasu, T., Kingsley, C., Dairkee, S., Meng, Z., Chew, K., Pinkel, D., Jain, A., Ljung, B., Esserman, L., Albertson, D., Waldman, F., and Gray, J. (2006). Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. *Cancer cell*, **10**, 529–541.

Chipman, H. A. and Gu, H. (2005). Interpretable dimension reduction. *Journal of Applied Statistics*, **32**, 969–987.

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A., Wodicka, T. G., Gabrielian, A. E., Landsma, D., Lockhar, D. J., and Davis, R. W. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. cell*, **2**, 65–73.

Cox, T. F. and Cox, M. A. A. (1994). *Multidimensional scaling*. Chapman, London.

Cureton, E. E. and Mulaik, S. A. (1975). The weighted varimax rotation and the promax rotation. *Psychometrika*, **40**, 183–195.

d'Aspremont, A., Ghaoui, L., Jordan, M., and Lanckriet, G. (2007). A direct formulation for sparse pca using semidefinite programming. *SIAM Review*, **49**, 434–448.

Everitt, B. (1974). *Cluster Analysis*. Heinemann Educational Books Ltd, London.

Everitt, B. S. and Dunn, G. (2001). *Applied Multivariate Data Analysis*. Arnold, London.

Farcomeni, A. (2009). An exact approach to sparse principal component analysis. *Computational Statistics*, **24**, 583–604.

Friedman, H. P. and Rubin, J. (1967). On some invariant criteria for grouping data. *J. Am. Statist. Ass.*, **62**, 1159–1178.

Gabriel, K. R. (1971). The biplot graphic display of matrices with application to principal component analysis. *Biometrika*, **58**, 453–467.

Gabriel, K. R. (2002). Goodness of fit of biplots and correspondence analysis. *Biometrika*, **89**, 423–436.

Gervini, D. and Rousson, V. (2004). Criteria for evaluating dimension-reducing components for multivariate data. *The American Statistician*, **58**, 72–76.

Gower, J. C. (1966). Some distance properties of latent root and vector methods used in multivariate analysis. *Biometrika*, **53**, 325–338.

Gower, J. C. (1967). Multivariate analysis and multidimensional geometry. *The Statistician*, **17**, 13–28.

Gower, J. C. and Hand, D. J. (1996). *Biplots*. Chapman & Hall, London.

Hand, D. J. (1981). Branch and bound in statistical data analysis. *The Statistician*, **30**, 1–13.

Harman, H. H. (1976). *Modern Factor Analysis*. Chicago University Press, Chicago.

Hartigan, J. A. (1975). *Clustering Algorithms*. Wiley, New York.

Hastie, T., Tibshirani, R., Eisen, M. B., Alizadeh, A., Levy, R., Staudt, L., Chan, W. C., Botstein, D., and Brown, P. (2000). 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, **1**, research0003.1–0003.21.

Hausman, R. E. (1982). Constrained multivariate analysis. In S. H. Zanakis and J. S. Rustagi, editors, *Optimization in statistics*, pages 137–151. North-Holland, Amsterdam.

Horn, R. A. and Johnson, C. A. (1985). *Matrix Analysis*. Cambridge University Press, Cambridge.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, **24**, 417–441, 498–520.

Jeffers, J. N. R. (1967). Two case studies in the application of principal component analysis. *Applied Statistics*, **16**, 225–236.

Jennrich, R. I. (1970). Orthogonal rotation algorithms. *Psychometrika*, **35**, 229–235.

Jennrich, R. I. (2001). A simple general procedure for orthogonal rotation. *Psychometrika*, **66**, 289–306.

Jennrich, R. I. (2004). Derivative free gradient projection algorithms for rotation. *Psychometrika*, **69**, 475–480.

Johnstone, I. M. and Lu, A. Y. (2004). Sparse principal components analysis (unpublished).

Jolliffe, I. T. (1972). Discarding variables in a principal component analysis i: Artificial data. *Applied statistics*, **21**, 160–173.

Jolliffe, I. T. (1973). Discarding variables in a principal component analysis ii: Real data. *Applied statistics*, **22**, 21–31.

Jolliffe, I. T. (1995). Rotation of principal components: Choice of normalization constraints. *Journal of Applied statistics*, **22**, 29–35.

Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer-verlag, New York.

Jolliffe, I. T. and Uddin, M. (2000). The simplified component technique: An alternative to rotated principal component. *Journal of Computational and Graphical Statistics*, **9**, 689–710.

Jolliffe, I. T., Trendafilov, N. T., and Uddin, M. (2003). A modified principal component technique based on the lasso. *Journal of Computational and Graphical Statistics*, **12**, 531–547.

Journée, M., Nesterov, Y., Richtárik, P., and Sepulchre, R. (2008). Generalized power method for sparse principal component analysis. Technical Report 2008/70. http://www.uclouvain.be/en-44508.html.

Kaiser, H. F. (1958). The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, **23**, 187–200.

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educ. Psychol. Meas.*, **20**, 141–151.

Knüsel, L. (2008). Chisquare as a rotation criterion in factor analysis. *Computational Statistics and Data Analysis*, **52**, 4243–4252.

Krzanowski, W. J. (1988). *Principles of Multivariate Analysis: A User's Perspective*. Oxford University Press, Oxford.

Lawley, D. N. and Maxwell, A. E. (1971). *Factor Analysis as a Statistical Method*. Butterworth, London.

Li, L. (2007). Sparse sufficient dimension reduction. *Biometrika*, **94**, 603–613.

Magnus, J. R. and Neudecker, H. (1988). Matrix differential calculus with application in statistics and econometrics.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1982). *Multivariate Analysis*. Academic Press, London.

MATLAB (2009). *MATLAB 2009a*. The Math Works, Inc.

McCabe, G. P. (1982). Principal variables. Technical Report 82-3, Purdue University.

McCabe, G. P. (1984). Principal variables. *Technometrics*, **26**, 137–144.

McLachlan, G. J., Do, K.-A., and Ambroise, C. (2004). *Analyzing Microarray Gene Expression Data*. Wiley, New Jersey.

Moghaddam, B., Weiss, Y., and Avidan, S. (2006). Spectral bounds for sparse pca: Exact and greedy algorithms. *Advances in Neural Information Processing Systems*, **18**, 915–922.

Mulaik, S. A. (1972). *The Foundations of Factor Analysis*. McGraw-Hill, New York.

Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *phil. Mag.*, **2**, 559–572.

Rand, W. M. (1971). Objective criteria for the evaluation of clustering methods. *Journal of the American Stat. Asso.*, **66**, 846–850.

Rao, C. R. (1964). The use and interpretation of principal component analysis in applied research. *Sankhya A*, **26**, 329–358.

Raymer, M. L., Punch, W. F., Goodman, E. D., Kuhn, L. A., and Jain, A. K. (2000). Dimensionality reduction using genetic algorithms. *IEEE Transactions on Evolutionary Computation*, **4**, 164–171.

Robert, P. and Escoufier, Y. (1976). A unifying tool for linear multivariate statistical methods: The rv-coefficient. *Applied Statistics*, **25**, 257–265.

Romesburg, H. C. (2004). *Cluster Analysis for Researchers*. Lulu Press, North Carolina.

Rousson, V. and Gasser, T. (2004). Simple component analysis. *Applied Statistics*, **53**, 539–555.

Seber, G. A. F. (2004). *Multivariate Observations*. Wiley, New Jersey.

Sherin, R. J. (1966). A matrix formulation of kaiser's varimax criterion. *Psychometrika*, **31**, 535 538.

Späth, H., editor (1980). *Cluster Analysis Algorithms for Data Reduction and Classification of Objects*. Ellis Horwood Ltd., Chichester.

ten Berge, J. M. F. (1984). A joint treatment of varimax and the problem of diagonalizing symmetric matrices simultaneously in the least squares sense. *Psychometrika*, **49**, 347–358.

Thalamuthu, A., Mukhopadhyay, I., Zheng, X., and Tseng, G. C. (2006). Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, **22**, 24052412.

Thurstone, L. L., editor (1947). *Multiple Factor Analysis*. University of Chicago Press, Chicago, IL.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of Royal Statistical Society*, **58**, 267–288.

Tibshirani, R., Walther, G., and Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *J. Royal Stat. Soc. B*, **63**, 411–423.

Trendafilov, N. T. (2006). The dynamical system approach to multivariate data analysis, a review. *Journal of Computational and Graphical Statistics*, **15**, 628–650.

Trendafilov, N. T. and Jolliffe, I. T. (2006). Projected gradient approach to the numerical solution of the scotlass. *Computational Statistics and Data Analysis*, **50**, 242–253.

Trendafilov, N. T. and Jolliffe, I. T. (2007). Dalass: Variable selection in discriminant analysis via the lasso. *Computational Statistics and Data Analysis*, **51**, 3718–3736.

Vichi, M. and Saporta, G. (2009). Clustering and disjoint principal component analysis. *Computational Statistics and Data Analysis*, **53**, 3194–3208.

Vigneau, E. and Qannari, E. M. (2003). Clustering of variables around latent components. *Communications in Statistics: Simulation and Computation*, **32**, 1131–1150.

Vines, S. K. (2000). Simple principal components. *Applied Statistics*, **49**, 441–451.

Webb, A. (1999). *Statistical Pattern Recognition*. Arnold, London.

Witten, D. M., Tibshirani, R., and Hastie, T. (2009). A penalized matrix decomposition, with applications to sparse principal components and canonical correlation. *Biostatistics*, **10**, 515–534.

Wood, M., Jolliffe, I. T., and Horgan, G. W. (2005). Variable selection for discriminant analysis of fish sounds using matrix correlations. *Journal of Agricultural, Biological, and Environmental Statistics*, **10**, 1–16.

Yeung, K. Y. and Ruzzo, W. L. (2001). Principal component analysis for clustering gene expression data. *Bioinformatics*, **17**, 763–774.

Zou, H., Hastie, T., and Tibshirani, R. (2006). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, **15**, 265–286.